NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

19960129 042 **THESIS**

DESIGN AND DEVELOPMENT OF A USER
INTERFACE AND USER MANUAL FOR THE
SOFTWARE MANAGEMENT FLIGHT SIMULATOR

by

Richard D. Davis, Jr.

September 1995

Thesis Advisor: Tarek Abdel-Hamid

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE DESIGN AND DEVELOPMENT OF A USER INTERFACE AND USER MANUAL FOR THE SOFTWARE MANAGEMENT FLIGHT SIMULATOR	5. FUNDING NUMBERS
6. AUTHOR(S) Davis, Jr., Richard D.	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT

The rapid growth of the software industry, and the complex projects undertaken, have made managing software development projects an increasingly difficult task. Despite the efforts of program managers, many projects are completed late and over-budget. This research effort designs a user-friendly Software Management Flight Simulator based on the Dynamica Model of Software Project Management. The purpose of the flight simulator is to allow managers to approach difficult project decisions and explore the impacts and interactions of key variables. The simulation model is the product of the prototype design approach and is written in Vensim, a graphic modeling and simulation program compatible with the IBM personal micro-computer. The Windows based graphic user interface provides an intuitive and flexible menu structure and represents the next generation of interactive gaming simulators based on the Dynamica Model.

14. SUBJECT TERMS System Dynamics, Simulations, Modeling, Software Engineering	15. NUMBER OF PAGES 90
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. 239-18 298-102

i

Approved for public release; distribution is unlimited.

**DESIGN AND DEVELOPMENT OF A USER
INTERFACE AND USER MANUAL FOR THE
SOFTWARE MANAGEMENT FLIGHT SIMULATOR**

Richard D. Davis, Jr.
Captain, United States Marine Corps.
B.S., The Citadel, 1995
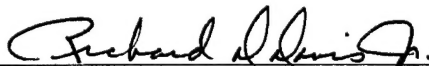
Submitted in partial fulfillment
of the requirements for the degree of

**MASTER OF SCIENCE IN INFORMATION TECHNOLOGY
MANAGEMENT**

from the
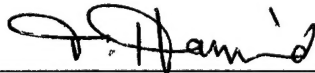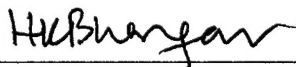
**NAVAL POSTGRADUATE SCHOOL
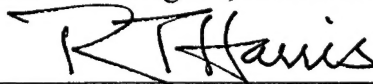September 1995**

Author: _____
Richard D. Davis, Jr.

Approved by: _____
Tarek Abdel-Hamid, Thesis Advisor

_____
Hemant Bhargava, Associate Advisor

_____
Reuben Harris, Chairman
Department of Systems Management

iii

ABSTRACT

The rapid growth of the software industry, and the complex projects undertaken, have made managing software development projects an increasingly difficult task. Despite the efforts of program managers, many projects are completed late and over-budget. This research effort designs a user-friendly Software Management Flight Simulator based on the Dynamica Model of Software Project Management. The purpose of the flight simulator is to allow managers to approach difficult project decisions and explore the impacts and interactions of key variables. The simulation model is the product of the prototype design approach and is written in Vensim, a graphic modeling and simulation program compatible with the IBM personal micro-computer. The Windows based graphic user interface provides an intuitive and flexible menu structure and represents the next generation of interactive gaming simulators based on the Dynamica Model.

TABLE OF CONTENTS

I. INTRODUCTION

A.   PROBLEM STATEMENT

In recent years, the development of software technology has progressed at an ever increasing rate. Boehm projects that computer software costs are expected to grow to more than $225 billion by 1995 in the U.S. and to more than $450 billion worldwide [Ref. 1]. Within the Department of Defense, billions of dollars are spent each year to develop software. Unfortunately, most projects are delayed, over budget, and often do not reflect the needs or desires of the users.

Software Engineering is the systematic application of sound engineering and management principles to create systems that meet the user's technical and economic objectives. Our increasingly interconnected and dynamic world challenges managers to find new ways to understand and control change. Intuition alone on the part of a software manager is not enough. Management simulators are becoming an important tool to create meaningful practice fields which accelerate the learning process.

The purpose of the Software Management Flight Simulator (SMFS) is to allow managers to approach particularly difficult decisions about software development issues and explore the various impacts and interactions of key variables. The SMFS provides a forum where managers can examine the different possible alternatives available, make a management decision, then advance several time periods in the future and see the effects of different courses of action. A well designed graphical user interface would enhance the effectiveness of the model as a learning aid and encourage users to further explore the potential possible with the Dynamica Model.

B.   PURPOSE OF RESEARCH

The purpose of this research is to convert the Dynamica Model from the Professional Dynamo modeling language to the Vensim modeling and simulation software and to design and develop an effective user interface and user manual. The interface design will significantly

increase the analytic capabilities of the model, provide a greater range of flexibility while running a simulation, and improve the overall quality of the learning experience for the user.

C.   SCOPE OF RESEARCH

The scope of the research is to design and development an effective user interface and user manual for the Dynamica model using the Vensim modeling and simulation software. Vensim is a Windows based decision support system which allows the user greater flexibility and a wide range of analytical tools which were unavailable in user interfaces developed using Professional Dynamo. This research focuses on the design considerations and the prototype development of a graphical user interface for a Software Management Flight Simulator. This research expands the earlier work completed by Captain C. E. Haury [Ref. 2], and Major D. W. Swindell [Ref. 3], in the Professional Dynamo interface.

D.   THESIS ORGANIZATION

Chapter II briefly discusses the role of management simulators and the factors which influence the user interface design. Chapter III addresses the system architecture and the capabilities of the Vensim modeling and simulation software. Chapter IV is the user's manual for the user interface.

II. MODELS AND SIMULATIONS

A.   FUNCTIONALITY

1. Management Flight Simulators

The decisions expected of a software project manager are dynamic and particularly difficult, because they often have indirect, delayed, nonlinear, and multiple feedback effects. Software tools must be developed which help managers conceptualize complex issues, design better operating policies and guide organization-wide learning [Ref. 4].

Simulators are designed to integrate the conceptualization, simulation, and decision-making process. Modeling simulators represent a "virtual world" where the pace of action can be varied. Actions that are irreversible in the real world become reversible. Complexity can be simplified. A virtual world compresses time and space, allowing managers to experience the long-term, system-wide consequences of decisions.

But to have an effective learning experience a manager must be tasked to do more than just witness a computer simulation. Experiments in the virtual world should lead to hypotheses which are tested through measurement and experimentation in the real world [Ref. 5]. The overall goals are to understand particular business and organizational dynamics and to develop systems thinking skills in problem identification, conceptualization, and analysis [Ref. 6].

In many respects, a software project manager is similar to a pilot in that both must control a system of great complexity and be prepared for the unexpected. No airline would send a pilot to fly an aircraft before they had trained extensively in a flight simulator on the ground. The simulator allows the pilot to learn, to make mistakes, and to experience the unexpected without risk to passengers or aircraft.

The Software Management Flight Simulator functions just as an aircraft simulator, it allows managers to make mistakes and to experience the unexpected without risk. During each simulated time period the software manager will receive feedback from past decisions and revise their estimates concerning project cost, project duration, requested staffing level, and the percent of the staff allocated to quality assurance.

3

The purpose of the simulation is to gain insight into the issues raised by the particular situation and to illustrate the difficulties of coordinating the operations and strategy as the manager of a software development project. The simulator helps managers see through superficial symptoms to the underlying causes of complex phenomena, reorganize perceptions into a clearer, more coherent picture of software dynamics. It is intended that managers learn and understand the dynamic relationships between the variables through the systematic exploration of the consequences of various strategies. [Ref. 7]

2. Human Factors and Goals

Before beginning the simulation, the managers should devise a strategy based on what they expect to happen. At the conclusion of the simulation, the actual results are compared to the expectations results. Discrepancies between the two figures can be examined further through causal tracing of the variables. The process of reflecting on discrepancies between expectations and outcomes establishes a discipline the managers then carry forward to experiments with new strategies. Without such discipline, simulation all too quickly becomes mere game playing. [Ref. 4]

A common problem with modeling simulations is that rather than conducting a series of controlled experiments, managers try a strategy and if it doesn't produce the desired outcome in a few time periods, they improvise. Instead of sticking with a strategy to see its long-term consequences, people quit a game which is going badly and start another [Ref. 9]. They behave the same way they do in real life, proceeding through trial and error, which produces little insight whether performance is good or bad. Treated as a game, simulations can reinforce misperceptions of feedback and result in cognitive errors in dynamic decision making [Ref. 4].

B.   SYSTEM DYNAMICS

The system dynamics philosophy is that organizations are viewed most effectively in terms of their common underlying flows instead of in terms of separate functions. The flows of people, money, materials, orders, and capital equipment, and the integrating flows of information can be identified in all organizations. It acts to dispel the component approach to organizations that promote interorganizational conflict and unrecognized suboptimization. A meaningful

4

systems framework results from tracing cause-and-effect chains through the relevant flow paths. [Ref. 10]

This system dynamics framework contains time lags and information feedback which can display complicated response patterns to relatively simple system or input changes. The behavior of such a system is complex and far beyond the capacity of intuition, and computer simulation is one of the most effective means available for supplementing and correcting human intuition. Edward Roberts identifies seven advantages associated with the construction of computer models by managerial policy makers:

1. Managers are required to complete a the rough mental sketch of the causes of the problem that they inevitably have in their heads.

2. During the process of formal model-building the managers discover and resolve various self-contradictions and ambiguities among their implicit assumptions about the problem.

3. Once the model is operating, the consequences of promising but tentative formulations are tested. Observation of model behavior gives rise to new hypotheses about the structure of the problem.

4. Once an acceptable standard of validity has been achieved formal experiments reveal the probable outcomes of many policy alternatives; novel policies may be discovered; "what if" situations can be explored.

5. An operating model is always complete, though in a sense never completed. Unlike many planning aids, which tend to provide assistance only at the moment the report is presented, a model is iterative. At any moment the model contains, in readily assessed form, the present best understanding of the problem.

6. Sensitivity analysis of the model reveals the areas in which genuine debate is needed and guides empirical investigation to important questions. If the true values of many parameters are unknown, the values that most affect model behavior need to be investigated first.

7. An operating model can be used to communicate with people who were not involved in building the model. By experimenting with changes in policies and model parameters and observing the effects of these changes on behavior, these people can be helped to better understand the dynamic forces at work in the real-world system. [Ref. 10]

Creating computer-based management tools involves first developing dynamic simulation models to explain critical aspects of the case history. However, if the models are to serve as a basis for a creative and active learning experience, rather than merely furnishing "the correct answer" it is necessary to design the means by which managers will interact with the model so as to reflect on their own assumptions and reasoning. This serves both for experimental determination of ideas related to policy improvement and for experimental testing of the proposed policy changes. [Ref. 6]

A systems dynamics model is not, as is sometimes supposed, a perfectly accurate representation of reality that can be trusted to make better decisions than people. It is a flexible tool that forces the people who use it to think harder and to confront one another, their common problems and themselves, directly and factually. Unlike the mental model, it is comprehensive, unambiguous, and subject to rigorous testing. [Ref. 10]

In general, system dynamics thinking is the body of skills, knowledge, and experience that enables a person to deal with complex dynamic systems. In particular, it is the ability to deal with phenomena that include dynamic behavior, feedback causality, time delays and non-linearities. One such model is the Dynamica Model of Software Project Management, which was developed by Tarek Abdel-Hamid while at the Massachusetts Institute of Technology.

C.   THE DYNAMICA MODEL

The goal of the Dynamica Model is to enhance our understanding of the software development process and to make predictions about the general process by which software systems are developed. The model is founded on the principles of System Dynamics, which emphasize the application of feedback loops and control systems principles [Ref. 11].

In the Dynamica Model, a causal loop structure (Figure 1) illustrates the circular relationships among cost/schedule estimates, workforce size, and productivity. These relationships provide a continuous simulation capability as the flows of information, resources, and products acting on a software project are modeled as numeric values as a function of time, rather than as analytical estimates [Ref. 11].
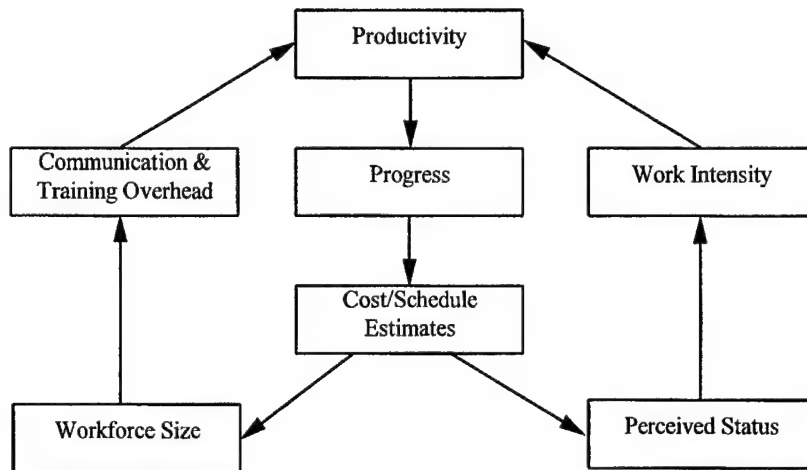
Figure 1. Circular relationship of Estimates, Workforce Size and Productivity [Ref. 11].

A major deficiency in much of the software development research has been the inability to integrate a knowledge of the components of the software development process in such a manner as to derive implications about the behavior the system as a whole. Problems persist because managers continue to believe that there are such things as unilateral causation, independent and dependent variables, origins, and terminations. While much attention has been placed on the phases and functions of the software development sequence, little attention has been focused on the whole lifecycle as an integral and continuous process [Ref. 12].

The Dynamica Model integrates the multiple functions of the software development process, including both the management functions (e.g. planning, control, staffing) as well as the software production-type activities (e.g. design, coding, reviewing, testing) [Ref. 13].

The model consists of four major subsystems. The Human Resource Management Subsystem comprises the hiring, training, assimilation, and transfer of the project's human resources. In the Planning Subsystem, initial project estimates are made to start the project, and

7

then those estimates are revised, when necessary, throughout the life of the project. The Software Production Subsystem allocates the available manpower among the different software production activities. The Controlling Subsystem compares the status of the project versus where it should be according to the manager's plan. Figure 2 provides an overview of the model's four subsystems and also illustrates the interrelatedness of the four subsystems. [Ref. 13]
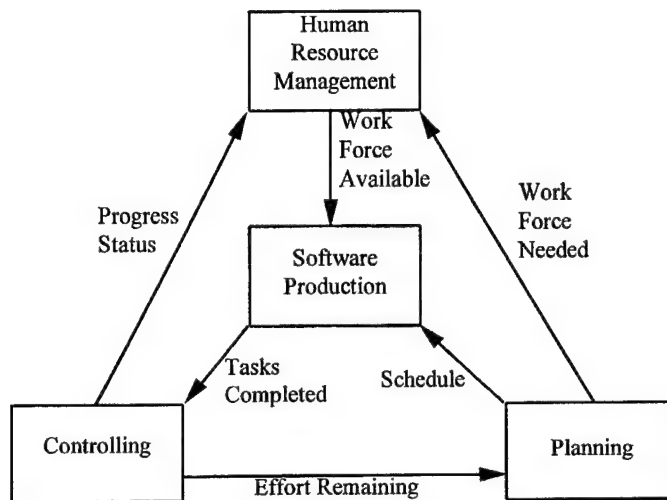
Figure 2. Software development subsystems from [Ref. 11].

D.   VENSIM SIMULATION ENVIRONMENT

1. Overview

Vensim is a Windows based visual modeling tool that allows the user to conceptualize, document, simulate, analyze and optimize a model or simulation. Models are constructed using the principles of causal loop and stock-flow diagrams to provide a simple and flexible method of building simulation models from diagrams consisting of words, arrows, valves and pipes.

The Vensim work environment is similar to many drawing programs in that it allows the user to create diagrams in a straightforward manner. Unlike drawing programs, however, the diagrams are interpreted by the program as having a meaning. Relationships between the variables are established by connecting the visual objects in the sketch tool with arrows which

are interpreted as unidirectional. The analysis tools treat arrows as indicating a direction of causality.

The program is divided functionally into three main components: the Workbench, the Sketch Tool, and the Equation Editor. The workbench is the main window from which Vensim is controlled. The workbench displays the main menu functions and also provides an input and output platform for building models and interacting with the analysis tools. The sketch tool is the primary building tool which allows the user to construct and change a model. For many applications, the sketch tool will be the only tool used. The equation editor allows the user to define equations for the variables in the model diagrams. [Ref. 15]

2. Advantages Over Professional Dynamo

Constructing a simulation using Vensim presents several advantages over similar models constructed using the simulation language Professional Dynamo. The most noteworthy of these advantages is a result of the visual presentation of the relationships between variables and flows. By integrating the equation generator with the sketch tool, Vensim provides an environment in which complex relationships can be modified as easily as connecting shapes using a drawing tool. This ability, together with several other key features, are the principle motivation for the transition of the Dynamica Model from Professional Dynamo to the Vensim. Listed below are some of the other enhancements available within the Vensim program.

- **Multiple Views:** Allows for more than one visual representation of the same logical model as a means of emphasizing different parts of the structure while tailoring the appearance to break up complicated models into simpler components.
- **Flexibility:** The visual and analytical presentation of a model can document situations and processes as they currently exist, or in the form of a "what if" analysis. This can provide an introduction to a functional area, a useful record of the evolution of ideas, or a means of determining how much change is really occurring in response to different initiatives.
- **Analyzing Problem Areas:** Once a simulation model has been constructed, simulations can be analyzed for reasons which explain the behavior of that model. The analysis tools make it simple to look at every assumption in a model in terms of

both structure and behavior. Every assumption and relationship in a model can be examined to develop a very thorough understanding of what is causing behavior.

- **Error Checking:** Vensim automatically adds arrows, as necessary, when moving from view to view to maintain causal consistency. In general, there may be a number of different perspectives for a model while maintaining equivalence among them. [Ref. 15]

III. FLIGHT SIMULATOR ARCHITECTURE

A.   FILE STRUCTURE

The files necessary to simulate the Dynamica Model were originally programmed using Professional Dynamo. The Professional Dynamo files were converted to the Vensim format using the DYN2VEN.EXE utility program. Although this utility is intended to provide a seamless conversion of the Professional Dynamo functions, several modifications to the basic (.vmf) file were required to correct equation errors and language specific functional inconsistencies. For example, the Vensim equivalent to the Dynamo PULSE function is the M_PULSE macro. All of the equations containing the PULSE function required a manual conversion to the Vensim format [Appendix A, Section VIII]. The converted Dynamica Model, restored to its previous working condition and annotated with a brief summary of the required modifications, is contained in Appendix A.

The Software Management Flight Simulator requires three component files to operate as an interactive simulation: the Vensim Model File (.vmf), the Vensim Custom Definition (.vcd), and the Vensim Graphic Definition (.vgd). These three files interact to form the simulation model gaming interface. A fourth file, the Vensim Data File (.vdf), is the product of a simulation which stores the values of each variable during the simulation. Combined, these four component files provide a flexible platform for the construction and analysis of simulation models and gaming interfaces. Figure 3 depicts the relationship of these component files.

11
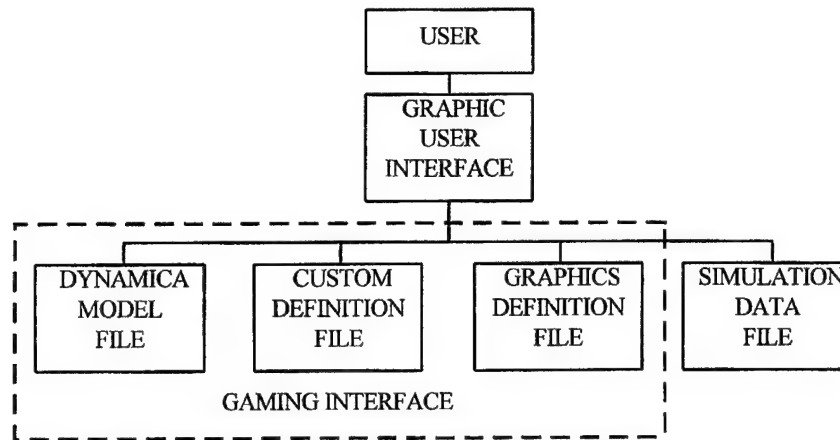
Figure 3. Software Management Flight Simulator Overview.

1. The Model File

The Software Management Flight Simulator is represented as a series of equations and causal relationships contained in the Vensim file with the extension (.vmf). Because the simulation model is the product of the DYN2VEN.EXE conversion utility, a graphic representation of the relationships between the variables is not available. The (.vmf) file may be viewed and modified as a plain text file from the Workbench using the View as Text option from the pull-down menu, however, the user should not attempt to modify the (.vmf) file. Interaction with the simulation model should be restricted to the gaming interface.

2. The Graphic Display

The gaming interface is created within the custom definition file (.vcd) as a script language, separated into individual "views" using the Vensim SCREEN function. The script language is a flexible tool which allows the designer to specify the size, position, color, and font of a text message; import predefined custom graphics and reports from the (.vgd) file, and modify the value of gaming variables. The screens are linked by a series of Shift to Screen commands which are executed in response to the input of the user. Combined, these individual views provide an interface which appears as a sequence of menu selections and graphic displays that allow the user navigate and interact with the simulation model.

12

3. The Custom Display

The predefined graphs and reports produced by Vensim are the result of the keywords, labels and variables contained in the graphic definition file (.vgd). This text file that contains each customized output as a specific named range which can be exported to the gaming interface and used to enhance the graphic displays presented in user interface.

4. The Data File

The data file is created using the run name specified by the user during the **Play New Game** option with the added extension (.vdf). The data file automatically stores the values of each variable, at every time interval, during the course of a simulation. However, the special format used by Vensim to archive data and simulation results is not directly portable to other applications. The data must be converted using the **Export Datasets** command from the **Datasets** pull-down menu from the Workbench. The export function allows the (.vdf) file to be formatted as a database, a spreadsheet, or a tab delimited text file which can then be imported by other applications.

B.   GAMING VARIABLES

To convert the basic Vensim model file to a interactive simulation the user must have the ability to modify the values of certain variables during the course of the simulation. Vensim requires that interactive variables be defined as GAME variables and assigned an initial value in the Vensim workbench **Datasets** Menu.

The transition of the Dynamica Model to the Software Management Flight Simulator required the following variables be designed as GAME variables: Percent of Manpower Allocated to Quality Assurance (FRMPQA), Project Duration (PROJDR), Project Cost (TOTMD1), and Staffing Level (WFS1). As gaming variables, the simulation will allow the user to input new values during each time interval.

C.   MENU STRUCTURE

The Software Management Flight Simulator presents the user with a graphical interface composed of a menu structure which is organized in a linear fashion. Figure 4 provides a block

13

diagram of the simulation model menu structure. The Main Menu offers the user three options: 1) **Play New Game** begins a simulation and creates a data file, 2) **Go Back** allows the user to return to the simulation model introduction, and 3) **Exit System** terminates the user interface.

Once the user begins a gaming simulation by selecting the **Play New Game** option, the user is presented the eight Home Screen menu selections (Figure 5). Seven of the options are arranged in a branch structure with each branch representing a series of customized reports, graphs and tables which return to the Home Screen along the original path. The eighth option, **End Simulation**, is an exception to the linear menu scheme in that it returns directly to the Main Menu. The advantage of this menu structure that it allows inexperienced computer users, or users unfamiliar with the simulation model, an easier interface to navigate.
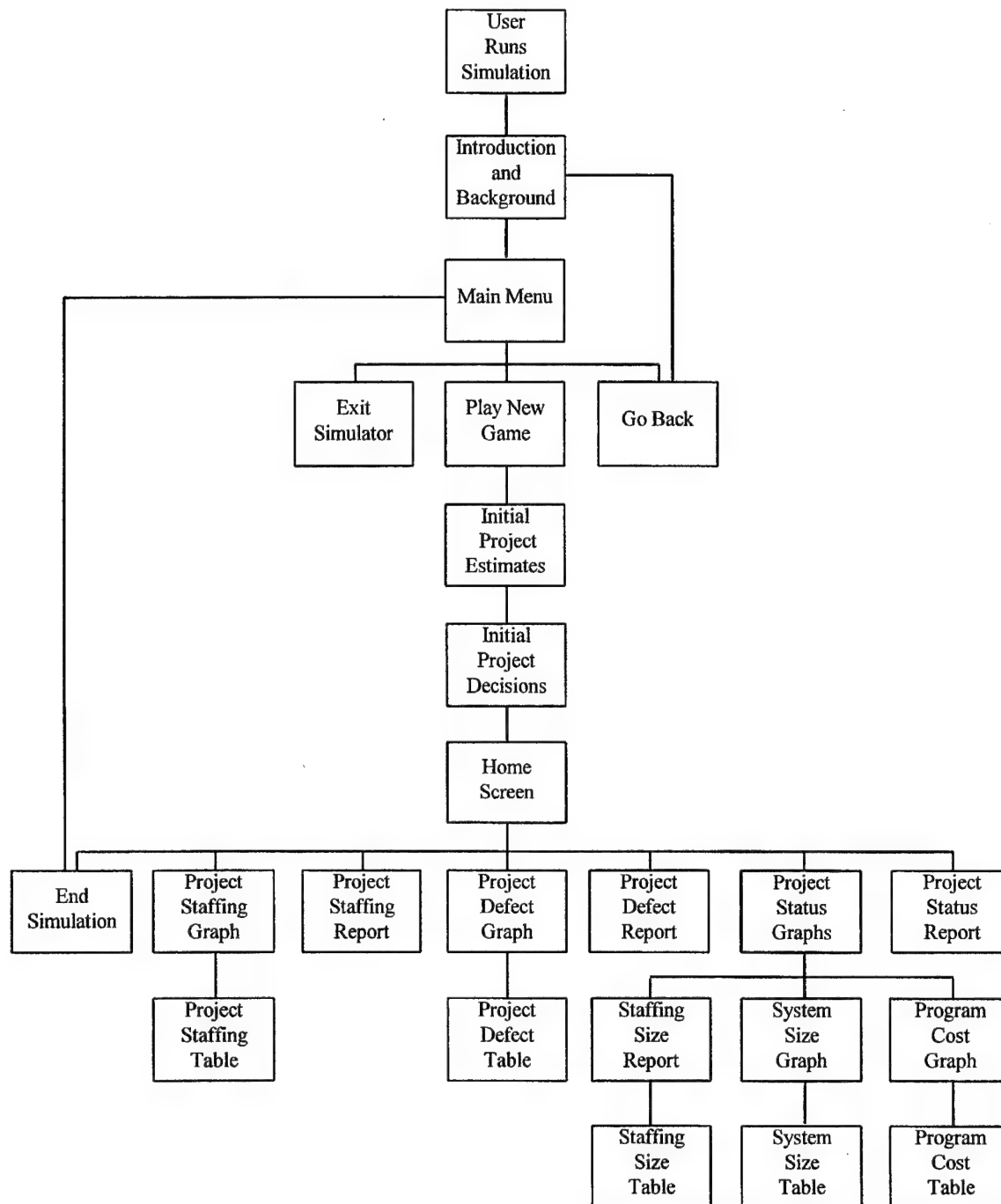
Figure 4. Simulation Model Menu Structure
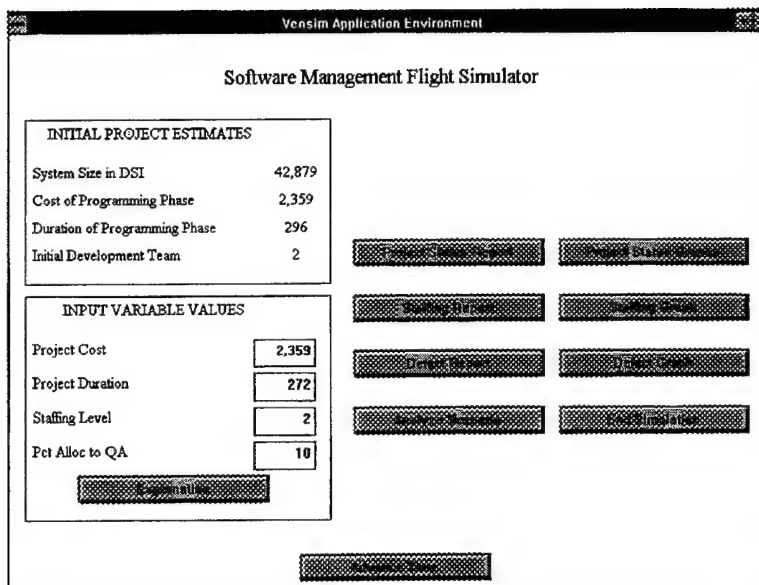
Figure 5. Gaming Interface Home Screen

IV. SOFTWARE USER'S MANUAL

A.   SYSTEM REQUIREMENTS

The Vensim program will run on any 286 or better PC running MS-DOS and Microsoft Windows 3.1. The minimum installation requires 1.6 megabytes of disk space, 6 megabytes for the full installation, and at least 2 megabytes of RAM memory. When working with large models, a 386 or better PC with a math coprocessor and 4 megabytes or more of memory is recommended. [Ref. 15]

B.   STARTING A SIMULATION

The Software Management Flight Simulator gaming interface may be activated by one of two methods. The first method requires the user start the Vensim program and **Open** the simulation model (.vmf) file from the File pull-down menu, then open the custom display (.vcd) file in the same manner. Finally, select the **Run Application Internally** option from the File pull-down menu to activate the gaming interface. The custom display (.vcd) file will not operate without the simulation model (.vmf) file active in the background. This method is used primarily as a convenience while editing files from the Workbench.

A second method to activate the gaming interface is to append the Vensim icon in the Windows Program Group to include the custom display (.vcd) file as a supplemental command. This method will activate the simulation model (.vmf) file and the gaming interface immediately with a double-click of the Vensim icon, bypassing the Workbench and avoiding any accidental modification of the simulation files.

Once the gaming interface is active, proceed through the simulation background information to the Main Menu and select the **Play New Game** option. A dialog box will appear and prompt the user for a file name to assign to the simulation data (.vdf) file. Type a new name or select a previous file to reuse. **Do not select the Base model as the data file for the simulation.** Vensim requires the Base file to remain unchanged for use during run comparisons while using the **Analyze Scenario** option.

Continue to follow the on-screen instructions and complete the initial program phase decisions, then proceed to the Home Screen (Figure 5). From the Home Screen, the user has

17

access to all of the pre-defined custom graphs, tables and reports. Analyze the situation of the software development project, determine the values of the decision variables and enter the figures in the blocks provided for: 1) Project Cost, 2) Project Duration, 3) Requested Staffing Level, and 3) Percent of Manpower to Allocate to Quality Assurance.

Click on the **Advance Time** button to advance to the next 40 day increment. Repeat this process until the Project Status Report indicates the Percent Delivered Source Instructions (DSI) Reported Complete equals 100% or the Game Over dialog box appears. At the completion of the simulation, select **OK** to close the dialog box, then select the **End Simulation** option to return to the Main Menu.

C.   ENHANCEMENTS TO THE GAMING INTERFACE

Converting the Software Management Flight Simulator from the Professional Dynamo language to Vensim has enhanced the overall appearance and functionality of the interface by presenting the information to the user in a more flexible and visually appealing Windows based format. The Vensim gaming interface offers the following capabilities not provided by the previous Professional Dynamo interface. The Vensim gaming interface offers the user the option of viewing the same graphs displayed in the Professional Dynamo interface, however, the Project Status Graph includes an **Unstack Graphs** option which separates the three variables into individual graphs. Additionally, each graph may be viewed as a table of values by selecting the **View Table** option, which displays the value of the variable at each time interval in the simulation.

The Windows based structure of the Vensim gaming interface provides the following features which allow the user to simultaneously view multiple reports, or reports from different time intervals:

- Windows can be sized to display more than one report at a time;
- Reports can be locked to the screen to prevent loss of the information when viewing graphs or selecting the **Advance Time** option;
- Pre-defined reports can be printed;
- Windows Clip and Paste features are available for report information; and

18

- Custom reports can be saved to a diskette.

The Vensim gaming interface allows the user greater flexibility for viewing the results of a simulation model. These features clarify the sometimes confusing displays of the previous interface and allow the user to focus on the data format most valued for a particular analysis and enhance the learning potential of the simulator.

D. NAVIGATING THE VENSIM WORKBENCH

The Vensim platform for developing and editing simulation models and gaming interfaces is the Workbench (Figure 6). The Workbench contains a comprehensive array of tools to create and edit the model files and facilitates the rapid design and prototyping of a simulation model and gaming interface.
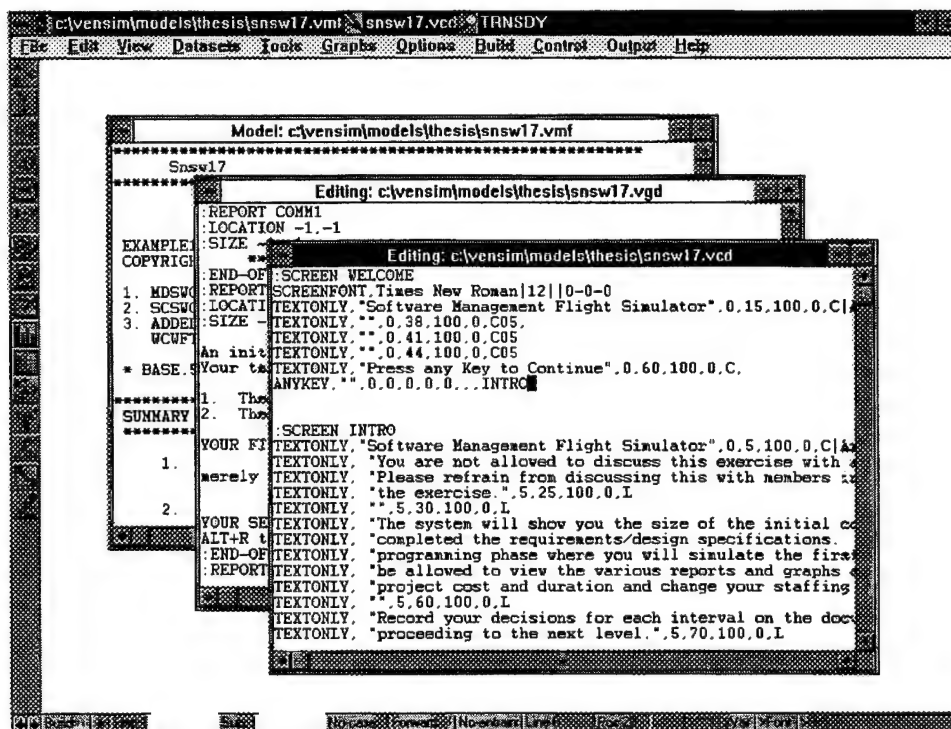
Figure 6. The Vensim Workbench

The following outline describes several of the most commonly used Workbench functions and provides a quick initial reference guide for future developers. Detailed

descriptions of each of the functions described below are available in the Vensim Reference Guide [Ref. 15] and the on-line help menu.

File Operations:

Check	Compiles the simulation model (.vmf) file and checks for syntax errors. Any change made to the simulation model (.vmf) file requires that the file be recompiled before execution.
Simulate	Executes the simulation model, without the gaming interface or time intervals, until the value of the control variables are exceeded and the simulation stops.

Dataset Operations:

Game	Permits the gaming of a simulation by activating a dialog box which displays the gaming variables and allows the user to modify these values before advancing to the next time interval. Simulation results can be viewed using any of the pre-defined graphs and reports contained in the **Custom Graphs** option of the **Graphs** pull-down menu.
Exporting Datasets	Converts a simulation data (.vdf) file to a database, spreadsheet, or tab delineated text file capable of being imported by another application.

Control:

Custom Graphs	Provides the user with quick and efficient dialog box to edit, create, display, copy, or delete the pre-defined graphs contained in the custom graphic (.vgf) file. The contents of the custom graphic (.vgf) file may also be edited as a text file from the Workbench.
Variable Selection	Displays a list of the defined variables within the simulation model (.vmf) file and allows the user to specify the active workbench variable (displayed in the center of the title bar). All of the analysis functions contained on the toolbar are applied to the active variable. A second method for selecting the active variable is to double-click on the variable name from either the sketch view or text view.

20

Datasets   Provides a means for the user to specify the active simulation data (.vdf) file for use with the causal tracing functions on the Workbench Toolbar. Multiple files may be selected when a direct comparison is desired between simulation data files.

The Workbench Toolbar:

The toolbar, located on the left side of the Workbench, provides the user with an assortment of useful functions for analyzing variables and simulation data (.vdf) files. The analysis functions are applied to the active variable, based on the loaded dataset (.vdf) file(s). The output appears as a Windows dialog box which can be printed or saved to diskette. The default toolbar includes the following functions:

- Causes Tree
- Causes Table
- Causes Strip Graph
- Strip Graph
- Bar Graph
- Units Check
- Gantt Chart
- Uses Tree
- Compare Runs
- Variable Loops
- Edit Equation

V. CONCLUSION

A.   ACCOMPLISHMENTS

Primary objective of this research was to convert the Dynamica Model from the Professional Dynamo modeling language to the Vensim modeling and simulation software, and to design and develop a graphic user interface for the simulation model.

The conversion of the Dynamica Model to Vensim required the modification of several variable equations, the conversion of the PULSE macro function, and the designation of specific gaming variables. The user interface was designed and developed using the principles of rapid prototyping. As a result of this iterative development process, several new custom graphs were added to the interface and the Vensim Causal Table function was used to add the presentation of a Tables of Values corresponding to each graph.

In accomplishing these objectives, the gaming interface provides the user with the following enhancements: 1) a menu structure that provides inexperienced computer users, or users new to the model, an intuitive and easy to follow information flow, 2) greater flexibility by providing the simulation data in multiple formats, and 3) a visually appealing Windows based appearance and improved functionality. The result of this effort is a simulation model which provides the user with more useful feedback and is easier to use.

B.   LESSONS LEARNED

First, the Vensim DYN2VEN.EXE utility program should have provided a seamless conversion of the Dynamica Model from Professional Dynamo to Vensim, however, this was not the case. It was discovered during the prototyping that several equations did not transfer correctly and a lengthy, and unanticipated, period of debugging the model file was necessary. An important factor to consider during prototyping is to test and verify that the source file is reliable before proceeding with the graphic interface. Also, despite the assurances provided in the software documentation, program conversions are not completely accurate and time should be allocated for such unscheduled debugging efforts.

Secondly, the development of visually appealing graphic displays is highly subjective among users and labor intensive to modify. One recommendation for Ventana Systems, Inc. is to include in future versions of the Vensim software, the ability to display an individual SCREEN from the custom display (.vcd) file to the Workbench for the purpose of facilitating the iterative design process. The present software version requires the developer to step through the gaming interface to the screen in question to view the results of the most recent change, then exit the simulation and return to Workbench for further modifications. This process is repeated until the desired effect is achieved.

Finally, though not identified in the User's Manual or the Reference Manual, Vensim initially reads the graphic display (.vgd) file when the model (.vmf) file is initially opened and does not reflect subsequent changes to the graphic display (.vgd) file unless the user exits the Vensim program and reenters the Workbench. Hours of development time were lost discovering this fact and adjusting the prototyping process because of it.

In summary, the prototyping design process of developing a simulation model and a gaming interface was successful in producing a product which met the research objectives, but the process was time consuming. The initial estimates for the duration of project phases were underestimated and the experience proved to be a valuable tool for learning the realities of software development.

C.   FUTURE DIRECTION

The Software Management Flight Simulator offers numerous areas for additional research in the following general categories:

Improved Functionality. One research objective which was examined during the course of this thesis, but was unable to be implement with the present version of Vensim, is the concept of replaying a previous simulation. The user would have the option of reloading a simulation data (.vdf) file, advancing the time interval to a key decision point, and replaying the simulation. The two outcomes would then be analyzed using the Vensim causal tracing functions. This function would greatly increase the learning potential of the flight simulator and is considered, at the present time, to be on the leading edge of simulation modeling.

Graphic Improvements. Vensim offers a highly flexible graphic display capability that can be customized to present a series of pop-up dialog boxes which could be dispersed throughout the gaming interface. The boxes could provide information such as:

Graphic representations of the Dynamica Model. The relationships between variables, presently in equation form, can easily be converted to feedback loops using the Vensim Workbench. These graphic representations can be displayed within the gaming interface to further enhance the user's understanding of the forces acting on a particular decision.

Descriptive summaries of the variables, simulation modules, and feedback loops could be included, in addition to a more extensive on-line help menu.

Reapply the modifications to the Dynamica Model, documented in Appendix A, using the simulation model (.vmf) file provided by Ventana System, Inc. The file contains equations based on the variables described using the long titles rather than the eight character Professional Dynamo variable names. Converting to the long titles will further enhance the readability of the graphs and the output of the analysis tools.

Upgrade to Vensim, Version 1.62, which has improved custom graphing capabilities. The upgrade enhances the document handling capabilities to view a graph within a pop-up dialog box and provides additional formatting options while displaying custom reports.

LIST OF REFERENCES

1. Boehm, B. W., "Improving Software Productivity." Computer, September 1987, p. 43-50.

2. Haury, Carson E., Design and Development of a User Interface for the Dynamica Model of Software Project Management, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1988.

3. Swindell, Daniel W., Design and Development of a User Interface for a System Dynamics Model of Software Management, Master's Thesis, Naval Postgraduate School, Monterey, California, March 1989.

4. Senge, Peter M. and Sterman, John D., "Systems Thinking and Organizational Learning: Acting Locally and Thinking Globally in the Organization of the Future." Transforming Organizations 1991.

5. Schon, D. The Reflective Practitioner. New York: Basic Books, 1983.

6. Graham, Alan K. and Senge, Peter M., "Computer-based case studies and learning laboratory projects." System Dynamics Review, Vol. 6 Number 1 (1990), p. 2.

7. Sterman, John D., "People Express Management Flight Simulator." Software and Briefing Book, 1988a.

8. Sterman, John D., "New M.B.A. students get oriented with Macs." MacWeek, 20 September 1988b.

9. Moissis, A., "Decision Making in the Insurance Industry: A dynamic Simulation Model and Experimental Results." Master's Thesis, Sloan School of Management, MIT, 1989.

10. Roberts, Edward B., "New Directions In Industrial Dynamics." Industrial Management Review, Fall 1964, p. 3-35.

11. Abdel-Hamid, Tarek K. and Madnick, Stuart E., Software Project Dynamics An Integrated Approach, New Jersey: Prentice Hall, 1991.

12. Jensen, R. W. and Tonies, C. C., Software Engineering. New Jersey: Prentice-Hall, 1979.

13. Abdel-Hamid, Tarek K. and Madnick, Stuart E., "On the Portability of Quantitive Software Estimation Models", Information & Management, 13 (1987), p. 1-10.

14. Abdel-Hamid, Tarek K., "Adapting, Correcting, and Perfecting Software Estimates: A Maintenance Metaphor." IEEE Software, March 1993, p. 20-29.

15. Ventana Systems, Inc., Vensim Reference Manual, Version 1.61, 1994.

APPENDIX A. VENSIM MODEL FORMAT FILE

```
*********************************************************
    SNSW15.VMF
*********************************************************~
```

EXAMPLE1 VERSION 1 JUNE 1991
COPYRIGHT (c) 1991, DYNAMICA

1. MDSWCH=0, TOTMD1=2359.60
2. SCSWCH=0, TDEV1=296.79
3. ADDED WCWFTP
    WCWFTP.K=SWITCH(HIREDY+ASIMDY,TMPRMR,TMPRMR)

* BASE.5 / BASE MODEL: VERSION 5

```
{*******************************************************
SUMMARY OF DYN2VEN CONVERSION MODIFICATIONS
*******************************************************
```

1.   CONVERTED WORK FORCE SOUGHT (WFS) EQUATION TO INSERT THE
CONSTANT (WFS1/ADMPPS) IN PLACE OF (WFNEED).

2.   CONVERTED WORK FORCE AT BEGINNING OF DESIGN (WFSTRT)
EQUATION TO INSERT THE CONSTANT VALUE (WFS1) IN PLACE OF
(INITIAL(TEAMSZ*INUDST)).

3.   CONVERTED DAILY MANPOWER ALLOCATED FOR QA (DMPQA)
EQUATION TO INSERT THE CONSTANT (FRMPQA) IN PLACE OF (AFMPQA).

4.   CONVERTED SCHEDULED COMPLETION DATE (SCHCDT) EQUATION TO
INSERT THE CONSTANT (PROJDR) IN PLACE OF (Integ((TIME STEP*(INDCDT-
SCHCDT)/SCHADT)/TIME STEP,TDEV)).

5.   CONVERTED TOTAL JOB SIZE IN MAN DATA (JBSZMD) TO INSERT THE
CONSTANT (TOTMD1) IN PLACE OF
(Integ((IRDVDT+IRTSDT+ARTJBM),DEVMD+TSTMD)).

6.   TOTMD1 EQUATION IS LISTED TWICE. REMOVED THE EQUATION
TOTMD1=944.

7.   INSERTED MISSING EQUATIONS FROM DYNAMO REPORTS. SEE SECTION VII.

8.   CHANGED (SAVEPER) AND (TIME_STEP) VALUES TO READ 40 AND 0.5 RESPECTIVELY.

9.   ADDED EQUATION (IPRJSZ=(RJBDSI)*(1-UNDEST)) WHICH WAS MISSING FROM VENSIM MODEL.

10.   CHANGED VARIABLES (TOTMD1),(PROJDR),(WFS1),(FRMPQA) TO "GAME" VARIABLES.

11.   ALL NON-MACRO TIME_STEP CHANGED TO TIME STEP.

12.   ADDED THE FOLLOWING EQUATIONS TO SUPPORT DYNAMO GRAPHS

CUMMD TD	PGM PERSON DAYS EXPENDED TO DATE
CMDSI KDSI	TOTAL KDSI COMPLETED
CMERD KDSI	DEFECT DENSITY
PJBSZT KDSI	ESTIMATED SYSTEM SIZE
PRQAMD PERIOD	QA PERSON DAYS PER KDSI DEVELOPED IN PERIOD
FRWFEX PCT	PERCENT OF WORKFORCE THAT IS EXPERIENCED

13.   CONVERTED EQUATIONS FOR VARIABLES (PRTKDV, TMSTOP, PRERD, AND PRQAMD) WHICH USE THE DYNAMO PULSE FUNCTION TO VENSIM M_PULSE FORMAT. SEE SECTION VIII.

14.   ADDED WFSINI AS INITIAL GAME VALUE. THIS PREVENTS THE CONSTANT UPDATE OF INITIAL DISPLAYED VALUES IN HOME_BASE SCREEN.

15.   CHANGED WFS1=GAME(2) TO WFS1=GAME(WFSINI) TO AVOID CONSTANT UPDATE IN HOME_BASE SCREEN DISPLAY.

16.   ADDED TDEVINI=INTEGER(TDEV1) TO AVOID DISPLAY OF DECIMAL FRACTION IN HOME_BASE SCREEN DISPLAY.

17.   ADDED TOTMD1INI AS INITIAL TOTAL ESTIMATED MAN DAYS.THIS PREVENTS THE CONSTANT UPDATE OF INITIAL DISPLAYED VALUES IN HOME_BASE SCREEN.

18.   CHANGED TOTMD1=GAME(2359.6) TO TOTMD1=GAME(TOTMD1INI) TO AVOID CONSTANT UPDATE IN HOME_BASE SCREEN DISPLAY.

```
{************************************************
I. HUMAN RESOURCE MANAGEMENT SUBSYSTEM
************************************************
}
WFNEW=INTEG((HIRERT-ASIMRT-NEWTRR),0)
   ~ PEOPLE
   ~ NEW WORKFORCE
   |

HIRERT=MAX(0,WFGAP/HIREDY)
   ~ PEOPLE/DAY
   ~ HIRING RATE
   |

HIREDY=40
   ~ DAYS
   ~ HIRING DELAY
   |

WFGAP=WFS-TOTWF
   ~ PEOPLE
   ~ WORKFORCE GAP
   |

NEWTRR=MIN(TRNFRT,WFNEW/TIME STEP)
   ~ PEOPLE/DAY
   ~. NEW EMPLOYEES TRANSFER RATE OUT
   |

TRNFRT=MAX(0,-WFGAP/TRNSDY)
   ~ PEOPLE/DAY
   ~ TRANSFER RATE OF PEOPLE OUT OF PROJECT
   |

TRNSDY=10
   ~ DAYS
   ~ TIME DELAY TO TRANSFER PEOPLE OUT
   |

ASIMRT=WFNEW/ASIMDY
   ~ PEOPLE/DAY
   ~ ASSIMILATION RATE OF NEW EMPLOYEES
   |
```

ASIMDY=80
   ~ DAYS
   ~ AVERAGE ASSIMILATION DELAY
   |

DMPTRN=WFNEW*TRPNHR
   ~ MAN DAYS/DAY
   ~ DAILY MANPOWER FOR TRAINING
   |

TRPNHR=0.2
   ~ DIMENSIONLESS
   ~ NUMBER OF TRAINERS PER NEW EMPLOYEE
   |

CMTRMD=INTEG(DMPTRN,0)
   ~
   ~ CUMULATIVE TRAINING MAN-DAYS
   |

WFEXP=INTEG((ASIMRT-EXPTRR-QUITRT),WFSTRT)
   ~ PEOPLE
   ~ EXPERIENCED WORKFORCE INITIAL VALUE OF EXPERIENCED WORKFORCE
LEVEL
   |

EXPTRR=MIN(WFEXP/TIME STEP,TRNFRT-NEWTRR)
   ~ PEOPLE/DAY
   ~ EXPERIENCED EMPLOYEES TRANSFER RATE
   |

QUITRT=WFEXP/AVEMPT
   ~ PEOPLE/DAY
   ~ EXPERIENCED EMPLOYEES QUIT RATE
   |

AVEMPT=673
   ~ DAYS
   ~ AVERAGE EMPLOYMENT TIME
   |

FTEXWF=WFEXP*ADMPPS
   ~ MEN
   ~ FULL-TIME-EQUIVALENT EXPERIENCED WF
   |

32

CELNWH=FTEXWF*MNHPXS
    ~ MEN
    ~ CEILING ON NEW HIREES
    |

MNHPXS=3
    ~ MEN/MEN
    ~ MOST NEW HIREES PER EXPERIENCED STAFF
    |

CELTWF=CELNWH+WFEXP
    ~ PEOPLE
    ~ CEILING ON TOTAL WORKFORCE
    |

WFS=MIN(CELTWF,WFS1/ADMPPS)
    ~ PEOPLE
    ~ WF SOUGHT
    |

WFSINI=2
    ~ PEOPLE
    ~ INITIAL VALUE OF STAFFING LEVEL
    |

WFS1=GAME(WFSINI)
    ~ PEOPLE
    ~ TOTAL REQUESTED STAFFING LEVEL
    |

TOTWF=WFNEW+WFEXP
    ~ PEOPLE
    ~ TOTAL WF LEVEL
    |

FTEQWF=TOTWF*ADMPPS
    ~ PERSONS
    ~ FULL TIME EQUIVALENT WF
    |

FRWFEX=WFEXP/TOTWF
    ~ DIMENSIOLESS
    ~ FRACTION OF WF THAT IS EXPERIENCED
    |

33

```
{****************************************
II. SOFTWARE PRODUCTION SUSBSYSTEM
****************************************


(A) MANPOWER ALLOCATION SECTOR


}
TOTDMP=TOTWF*ADMPPS
   ~ MAN-DAYS/DAY
   ~ TOTAL DAILY MANPOWER
   |

ADMPPS=1
   ~ DAY/DAY
   ~ AVERAGE DAILY MANPOWER PER STAFF
   |

CUMMD=INTEG(TOTDMP,0.0001)
   ~ MAN DAYS
   ~ CUMULATIVE MAN-DAYS EXPENDED
   |

DMPATR=TOTDMP-DMPTRN
   ~ MAN-DAYS/DAY
   ~ DAILY MANPOWER AVAILABLE AFTER TRAINING
   |

AFMPQA=ACTIVE INITIAL(PFMPQA*(1+ADJQA),PFMPQA)
   ~ DIMENSIONLESS
   ~ ACTUAL FRCATION OF MANPOWER FOR QA
   |

QO=0
   ~
   ~ QUALITY OBJECTIVE ... NORMAL QO = 0
   |

PFMPQA=TPFMQA(PJBAWK)*(1+QO/100)
   ~ DIMENSIONLESS
   ~ PLANNED FRACTION OF MANPOWER FOR QA
   |
```

34

TPFMQA(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0.15,0)
    ~
    ~ |

ADJQA=TADJQA(SCHPR)
    ~ Percent
    ~ Percent ADJUSTMENT IN PFMPQA
    |

TADJQA(0,0.1,0.2,0.3,0.4,0.5,
0,-0.025,-0.15,-0.35,-0.475,-0.5)
    ~
    ~ |

DMPQA=MIN(((FRMPQA/100)*TOTDMP),0.9*DMPATR)
    ~ MAN-DAYS/DAY
    ~ DAILY MANPOWER ALLOCATED FOR QA
    |

FRMPQA=GAME(10)
    ~ PERSONS
    ~ FRACTION OF MANPOWER ALLOCATED FOR QA
    |

CMQAMD=INTEG(DMPQA,0)
    ~ MAN-DAYS
    ~ CUMULATIVE QA MAN-DAYS
    |

DMPSWP=DMPATR-DMPQA
    ~ MAN-DAYS/DAY
    ~ DAILY MANPOWER FOR SOFTWARE PRODUCTION
    |

DESECR=ACTIVE INITIAL(DTCERR/DESRWD,0)
    ~ ERRORS/DAY
    ~ DESIRED ERROR CORRECTION RATE
    |

DESRWD=15
    ~ DAYS
    ~ DESIRED REWORK DELAY
    |

35

DMPRW=ACTIVE INITIAL(MIN((DESECR*PRWMPE),DMPSWP),0)
   ~ MAN-DAYS/DAY
   ~ DAILY MANPOWER ALLOCATED FOR REWORK
   |

PRWMPE=INTEG((RWMPPE-PRWMPE)/TARMPE,0.5)
   ~ MAN-DAYS/ERROR
   ~ PERCEIVED REWORK MANPOWER NEEDED PER ERROR
   |

TARMPE=10
   ~ DAYS
   ~ TIME TO ADJUST PRWMPE
   |

CMRWMD=INTEG(DMPRW,0)
   ~ MAN DAYS
   ~ CUMULATIVE REWORK MAN-DAYS
   |

DMPDVT=DMPSWP-DMPRW
   ~ MAN-DAYS/DAYS
   ~ DAILY MANPOWER FOR DEVELOPMENT/ TESTING
   |

CMDVMD=INTEG((TIME STEP*DMPDVT*(1-FREFTS))/TIME STEP,0)
   ~ MAN DAYS
   ~ CUMULATVE DEVELOPMENT MAN-DAYS
   |

{
        (B) SOFTWARE DEVELOPMENT SECTOR

}
SDVRT=ACTIVE INITIAL(MIN((DMPSDV*SDVPRD),TSKPRM/TIME STEP),0)
   ~ TASKS/DAY
   ~ SOFTWARE DEVELOPMENT RATE
   |

DMPSDV=DMPDVT*(1-FREFTS)
   ~ MAN-DAYS/DAY
   ~ DAILY MANPOWER FOR SOFTWARE DEVELOPMENT
   |

FREFTS=TFEFTS(TSKPRM/PJBSZ)
    ~ DIMENSIONLESS
    ~ FRACTION OF EFFORT FOR SYSTEM TESTING
    |

TFEFTS(0,0.04,0.08,0.12,0.16,0.2,
1,0.5,0.28,0.15,0.05,0)
    ~
    ~ |

SDVPRD=POTPRD*MPDMCL
    ~ TASKS/MAN-DAY
    ~ SOFTWARE DEVELOPMENT PRODUCTIVITY
    |

POTPRD=ANPPRD*MPPTPD
    ~ TASKS/MAN-DAY
    ~ POTENTIAL PRODUCTIVTY
    |

ANPPRD=FRWFEX*NPWPEX+(1-FRWFEX)*NPWPNE
    ~ TASKS/MAN-DAY
    ~ AVERAGE NOMINAL POTENTIAL PRODUCTIVITY
    |

NPWPEX=1
    ~ TSK/M-D
    ~ NOMINAL POTENTIAL PRODUCTIVITY OF EXP EMPLOYEE
    |

NPWPNE=0.5
    ~ TSK/M-D
    ~ NOMINAL POTENTIAL PROD OF NEW EMPL.
    |

MPPTPD=TMPTPD(PJBAWK)
    ~
    ~ MULTIPLIER TO POTENTIAL PRODUCTIVITY DUE TO LEARNING
(DIMENSIONLESS
    |

TMPTPD(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
1,1.0125,1.0325,1.055,1.09,1.15,1.2,1.22,1.245,1.25,1.25)
    ~
    ~ |

MPDMCL=AFMDPJ*(1-COMMOH)
    ~
    ~ MULTIPLIER TO PRODUCTIVITY DUE TO MOTIVATION & COMM LOSSES
(DIMENSI
    |

COMMOH=TCOMOH(TOTWF)
    ~ DIMENSIONLESS
    ~ COMMUNICATION OVERHEAD
    |

TCOMOH(0,5,10,15,20,25,30,
0,0.015,0.06,0.135,0.24,0.375,0.54)
    ~
    ~ |

NFMDPJ=0.6
    ~ DIMENSIONLESS
    ~ NOMINAL FRACTION OF A MAN-DAY ON PROJECT
    |

AFMDPJ=INTEG(WRADJR,NFMDPJ)
    ~ DIMENSIONLESS
    ~ ACTUAL FRACTION OF A MAN-DAY ON PROJECT
    |

WRADJR=(WKRTS-AFMDPJ)/WKRADY
    ~ 1/DAY
    ~ WORK RATE ADJUSTMENT RATE
    |

WKRADY=NWRADY*EWKRTS
    ~ DAYS
    ~ WORK RATE ADJUSTMENT DELAY
    |

NWRADY=TNWRAD(TIMERM)
    ~ DAYS
    ~ NORMAL WORK RATE ADJUSTMENT DELAY
    |

TNWRAD(0,5,10,15,20,25,30,
2,3.5,5,6.5,8,9.5,10)
    ~
    ~ |

EWKRTS=IF THEN ELSE(WKRTS >= AFMDPJ,1,0.75)
   ~ DIMENSIONLESS
   ~ EFFECT OF WORK RATE SOUGHT
   |

WKRTS=(1+PBWKRS)*NFMDPJ
   ~ DIMENSIONLESS
   ~ WORK RATE SOUGHT
   |

MAXMHR=INITIAL(1)
   ~ DIMENSIONLESS
   ~ MAXIMUM BOOST IN MAN-HOURS
   |

PBWKRS=IF THEN ELSE(PMDSHR >=
0,(MDHDL/(FTEQWF*(OVWDTH+0.0001))),(MDHDL/(TMDPSN-MDHDL
+0.0001)))
   ~ Percent
   ~ Percent BOOST IN WORK RATE SOUGHT
   |

MDHDL=IF THEN ELSE(PMDSHR >= 0,MIN(MAXSHR,PMDSHR),-EXSABS)*CTRLSW
   ~ MAN-DAYS
   ~ MAN-DAYS THAT WILL BE HANDLED OR ABSORBED
   |

CTRLSW=1
   ~ Zero or One
   ~ CONTROL SWITCH ... ALLOWS US TO TEST POLICY OF NO OVERWORK
   |

EXSABS=MAX(0,(TEXABS(TMDPSN/MDRM)*MDRM-TMDPSN))
   ~ MAN-DAYS
   ~ MAN-DAY EXCESSES THAT WILL BE ABSOBED
   |

TEXABS(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
0,0.2,0.4,0.55,0.7,0.8,0.9,0.95,1,1,1)
   ~
   ~ |

MAXSHR=(OVWDTH*FTEQWF*MAXMHR)*WTOVWK
   ~ MAN-DAYS
   ~ MAXIMUM SHORTAGE IN MAN-DAYS THAT CAN BE HANDLED
   |

WTOVWK=IF THEN ELSE(Time >= BRKDTM+RLXTMC,1,0)
    ~ Zero or One
    ~ WILLINGNESS TO OVERWORK
    |

BRKDTM=INTEG((TIME STEP*(1/TIME STEP)*(MAX(BRKDTM,IF THEN
ELSE(OVWDTH = 0,(Time+TIME STEP
),0))-BRKDTM))/TIME STEP,-1)
    ~
    ~ TIME OF LAST EXHAUSTION BREAKDOWN
    |

RLXTMC=INTEG(((IF THEN ELSE(EXHLEV/MXEXHT >= 0.1,1,-RLXTMC/TIME STEP)-
((1/TIME STEP
)*RLXTMC*IF THEN ELSE(OVWDTH = 0,1,0))),0)
    ~
    ~ VARIABLE THAT CONTROLS TIME TO DE-EXHAUST
    |

OVWDTH=NOVWDT*MODTEX
    ~ DAYS
    ~ OVERWORK DURATION THRESHOLD
    |

NOVWDT=TNOWDT(TIMERM)
    ~ DAYS
    ~ NOMINAL OVERWORK DURATION THRESHOLD
    |

TNOWDT(0,10,20,30,40,50,
0,10,20,30,40,50)
    ~
    ~ |

MODTEX=TMODEX(EXHLEV/MXEXHT)
    ~ DIMENSIONLESS
    ~ EFFECT OF EXHAUSTION ON OVERWORK DURATION THRESHOLD
    |

TMODEX(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
1,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0.1,0)
    ~
    ~ |

EXHLEV=INTEG((RIEXHL-RDEXHL),0)
~ EXHAUST UNITS
~ EXHAUSTION LEVEL
|

RIEXHL=TRIXHL((1-AFMDPJ)/(1-NFMDPJ))
~ EXHAUST UNITS/DAY
~ RATE OF INCREASE IN EXHAUSTION LEVEL
|

TRIXHL(-0.5,-0.4,-0.3,-0.2,-0.1,7.45058e-009,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,
1,
2.5,2.2,1.9,1.6,1.3,1.15,0.9,0.8,0.7,0.6,0.5,0.4,0.3,0.2,0,0)
~
~ |

RDEXHL=IF THEN ELSE(0 >= RIEXHL,EXHLEV/EXHDDY,0)
~ EXHAUST UNITS/DAY
~ RATE OF DEPLETION IN EXHAUSTION LEVEL
|

EXHDDY=20
~ DAYS
~ EXHAUSTION DEPLETION DELAY TIME
|

MXEXHT=50
~ EXHAUST UNITS
~ MAXIMUM TOLERABLE EXHAUSTION
|

{
        (C) QUALITY ASSURANCE AND REWORK SECTOR

}
QART=DELAY3(SDVRT,AQADLY)
~ TASKS/DAY
~ FOR QA RATE
|

TSKWK=INTEG((SDVRT-QART),0)
~ TASKS
~ TASKS WORKED
|

AQADLY=10
  ~ DAYS
  ~ AVERAGE DELAY FOR QA
  |

CUMTQA=INTEG((QART-TSRATE),0)
  ~ TASKS
  ~ CUMULATIVE TASKS QA'ED
  |

ANERPT=MAX(PTDTER/(TSKWK+0.0001),0)
  ~ ERRORS/TASK
  ~ AVERAGE # OF ERRORS PER TASK
  |

QAMPNE=NQAMPE*(1/MPDMCL)*MDEFED
  ~ MAN-DAYS/ERROR
  ~ QA MANPOWER NEEDED TO DETECT AVERAGE ERROR
  |

NQAMPE=TNQAPE(PJBAWK)
  ~ MAN-DAYS/ERROR
  ~ NOMINAL QA MANPOWER NEEDED TO DETECT AVERAGE ERROR
  |

TNQAPE(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
0.4,0.4,0.39,0.375,0.35,0.3,0.25,0.225,0.21,0.2,0.2)
  ~
  ~ |

MDEFED=TMDFED(ERRDSY)
  ~ DIMENSIONLESS
  ~ MULTIPLIER TO DETECTION EFFORT DUE TO ERROR DENSITY
  |

TMDFED(0,1,2,3,4,5,6,7,8,9,10,
50,36,26,17.5,10,4,1.75,1.2,1,1,1)
  ~
  ~ |

ERRDSY=ANERPT*1000/DSIPTK
  ~ ERRORS/KDSI
  ~ ERROR DENSITY
  |

PERDRT=DMPQA/QAMPNE
  ~ ERRORS/DAY
  ~ POTENTIAL ERROR DETECTION RATE
  |

ERRDRT=MIN(PERDRT,PTDTER/TIME STEP)
  ~ ERRORS/DAY
  ~ ERROR DETECTION RATE
  |

CMERD=INTEG(ERRDRT,0)
  ~ ERRORS
  ~ CUMULATIVE ERRORS DETECTED
  |

PRCTDT=100*CMERD/(CUMERG+0.001)
  ~ Percent
  ~ PERCENT ERRORS DETECTED
  |

ERRSRT=QART*ANERPT
  ~ ERRORS/DAY
  ~ ERROR ESCAPE RATE
  |

CMERES=INTEG(ERRSRT,0)
  ~ ERRORS
  ~ CUMULATIVE ERRORS THAT ESCAPED
  |

PTDTER=INTEG((ERRGRT-ERRDRT-ERRSRT),0)
  ~ ERRORS
  ~ POTENTIALLY DETECTABLE ERRORS
  |

ERRGRT=SDVRT*ERRPTK
  ~ ERRORS/DAY
  ~ ERROR GENERATION RATE
  |

ERRPTK=NERPTK*MERGSP*MERGWM
  ~ ERRORS/TASK
  ~ ERRORS PER TASK
  |

NERPTK=NERPK*DSIPTK/1000
   ~ ERRORS/TASK
   ~ NOMINAL # OF ERRORS COMMITTED PER TASK
   |

NERPK=TNERPK(PJBAWK)
   ~ ERRORS/KDSI
   ~ NOMINAL # OF ERRORS COMMITTED PER KDSI
   |

TNERPK(0,0.2,0.4,0.6,0.8,1,
25,23.86,21.59,15.9,13.6,12.5)
   ~
   ~ |

MERGSP=TMEGSP(SCHPR)
   ~
   ~ MULTIPLIER TO ERROR GENERATION DUE TO SCHEDULE PRESSURE
(DIMENSIONL
   |

TMEGSP(-0.4,-0.2,0,0.2,0.4,0.6,0.8,1,
0.9,0.94,1,1.05,1.14,1.24,1.36,1.5)
   ~
   ~ |

MERGWM=TMEGWM(FRWFEX)
   ~ DIMENSIONLESS
   ~ MULTIPLIER TO ERROR GENERATION DUE TO WORKFORCE MIX
   |

TMEGWM(0,0.2,0.4,0.6,0.8,1,
2,1.8,1.6,1.4,1.2,1)
   ~
   ~ |

CUMERG=INTEG(ERRGRT,0)
   ~ ERRORS
   ~ CUMULATIVE ERRORS GENERATED DIRECTLY DURING WORKING
   |

DTCERR=INTEG((ERRDRT-RWRATE),0)
   ~ ERRORS
   ~ DETECTED ERRORS
   |

44

RWRATE=DMPRW/RWMPPE
   ~ ERRORS/DAY
   ~ REWORK RATE
   |

RWMPPE=NRWMPE/MPDMCL
   ~ MAN-DAYS/ERROR
   ~ REWORK MANPOWER NEEDED PER ERROR
   |

NRWMPE=TNRWME(PJBAWK)
   ~ MAN-DAYS/ERROR
   ~ NOMINAL REWORK MANPOWER NEEDED PER ERROR
   |

TNRWME(0,0.2,0.4,0.6,0.8,1,
0.6,0.575,0.5,0.4,0.325,0.3)
   ~
   ~ |

CMRWED=INTEG(RWRATE,0)
   ~ ERRORS
   ~ CUMULATIVE REWORKED ERRORS DURING DEVELOPMENT
   |

{
        (D) SYSTEM TESTING SECTOR

}
UDAVER=INTEG((AEGRT+AERGRT-AERRRT-DCRTAE),0)
   ~ ERRORS
   ~ UNDETECTED ACTIVE ERRORS
   |

AEGRT=(ERRSRT+BDFXGR)*FRAERR
   ~ ERRORS/DAY
   ~ ACIVE ERRORS GENERATION RATE
   |

BDFXGR=RWRATE*PBADFX
   ~ ERRORS/DAY
   ~ BAD FIXES GENERATE RATE
   |

45

PBADFX=0.075
    ~ FRACTION
    ~ PERCENT BAD FIXES
    |

FRAERR=TFRAER(PJBAWK)
    ~ DIMENSIONLESS
    ~ FRACTION OF ESCAPING ERRORS THAT WILL BE ACTIVE
    |

TFRAER(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
1,1,1,1,0.95,0.85,0.5,0.2,0.075,0,0)
    ~
    ~ |

AERGRT=SDVRT*SMOOTH(AERRDS,TSAEDS)*MAERED
    ~ ERRORS/DAY
    ~ ACTIVE ERRORS REGENERATION RATE
    |

MAERED=TMERED(SMOOTH(AERRDS*1000/DSIPTK,TSAEDS))
    ~
    ~ MULTIPLIER TO ACTIVE ERROR REGENERATION DUE TO ERROR DENSITY
(DIMEN
    |

TMERED(0,10,20,30,40,50,60,70,80,90,100,
1,1.1,1.2,1.325,1.45,1.6,2,2.5,3.25,4.35,6)
    ~
    ~ |

TSAEDS=40
    ~ DAYS
    ~ TIME TO SMOOTH ACTIVE ERROR DENSITY (AERRDS)
    |

AERRDS=UDAVER/(CUMTQA+0.1)
    ~ ERRORS/TASK
    ~ ACTIVE ERROR DENSITY
    |

AERRRT=UDAVER*AERRFR
    ~ ERRORS/DAY
    ~ ACTIVE ERRORS RETIRING RATE
    |

AERRFR=TERMFR(PJBAWK)
  ~ 1/DAYS
  ~ ACTIVE ERRORS RETIRING FRACTION
  |

TERMFR(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
0,0,0,0,0.01,0.02,0.03,0.04,0.1,0.3,1)
  ~
  ~ |

DCRTAE=MIN(TSRATE*AERRDS,UDAVER/TIME STEP)
  ~ ERRORS/DAY
  ~ DETECTION/CORRECTION RATE OF ACTIVE ERRORS
  |

UDPVER=INTEG((PEGRT+AERRRT-DCRTPE),0)
  ~ ERRORS
  ~ UNDETECTED PASSIVE ERRORS
  |

PEGRT=(ERRSRT+BDFXGR)*(1-FRAERR)
  ~ ERRORS/DAY
  ~ PASSIVE ERRORS GENERATION RATE
  |

DCRTPE=MIN(TSRATE*PERRDS,UDPVER/TIME STEP)
  ~ ERRORS/DAY
  ~ DETECT/CORRECT RATE OF PASSIVE ERRORS
  |

CMRWET=INTEG((DCRTPE+DCRTAE),0)
  ~ ERRORS
  ~ CUMULATIVE ERRORS REWORKED IN TESTING PHASE
  |

ALESER=UDAVER+UDPVER+CMRWET
  ~ ERRORS
  ~ ALL ERRORS THAT ESCAPED AND WERE GENERATED
  |

DMPTST=DMPDVT*FREFTS
  ~ MAN DAYS/DAY
  ~ DAILY MANPOWER FOR TESTING
  |

47

CMTSMD=INTEG(DMPTST,0)
~ MAN DAYS
~ CUMULATIVE TESTING MAN-DAYS
|

TSRATE=MIN(CUMTQA/TIME STEP,DMPTST/TMPNPT)
~ TASKS/DAY
~ TESTING RATE
|

TMPNPT=(TSTOVH*(DSIPTK/1000)+TMPNPE*(PERRDS+AERRDS))/MPDMCL
~ MAN-DAYS/TASK
~ TESTING MANPOWER NEEDED PER TASK
|

TSTOVH=1
~ MAN-DAYS/KDSI
~ TESTING EFFORT OVERHEAD
|

TMPNPE=0.15
~ MAN-DAY/ERROR
~ TESTING MANPOWER NEEDED PER ERROR
|

PTKTST=CUMTKT/PJBSZ
~ Percent
~ Percent OF TASKS TESTED
|

PERRDS=UDPVER/(CUMTQA+0.0001)
~ ERRORS/TASK
~ PASSIVE ERROR DENSITY
|

CUMTKT=INTEG(TSRATE,0)
~ TASKS
~ CUMULATIVE TASKS TESTED
|

ALLERR=PTDTER+DTCERR+CMRWED+UDAVER+UDPVER+CMRWET
~ ERRORS
~ ALL ERRORS
|

48

ALLRWK=CMRWED+CMRWET
   ~ ERRORS
   ~ ALL ERRORS REWORKED ... IN DEVELOPMENT AND TESTING
   |

{*************************
III. CONTROL SUBSYSTEM
****
}
CMTKDV=INTEG(SDVRT,0)
   ~ TASKS
   ~ CUMULATIVE TASKS DEVELOPED
   |

PJBAWK=CMTKDV/RJBSZ
   ~ Percent
   ~ Percent OF JOB ACTUALLY WORKED
   |

PJDPRD=TSKPRM/(MDPRNT+0.1)
   ~ TASKS/MAN-DAY
   ~ PROJECTED DEVELOPMENT PRODUCTIVITY
   |

MDPRNT=MAX(0,MDRM-MDPNRW-MDPNTS)
   ~ MAN-DAYS
   ~. MAN DAYS PERCEIVED REMAINING FOR NEW TASKS
   |

MDPNRW=DTCERR*PRWMPE
   ~
   ~ MAN DAYS PERCEIVED NEEDED FOR REWORKING ALREADY DETECTED
ERRORS (MD
   |

ASSPRD=PJDPRD*WTPJDP+PRDPRD*(1-WTPJDP)
   ~ TASKS/MAN-DAY
   ~ ASSUMED PRODUCTIVITY
   |

PRDPRD=CMTKDV/(CUMMD-CMTSMD)
   ~ TASKS/MAN-DAY
   ~ PERCEIVED DEVELOPMENT PRODUCTIVITY
   |

WTPJDP=MPWDEV*MPWREX
   ~ DIMENSIONLESS
   ~ WEIGHT TO PROJECTED DEVELOPMENT PRODUCTIVITY
   |

MPWDEV=TMPDEV(PJBPWK/100)
   ~ DIMENSIONLESS
   ~ MULTIPLIER TO PRODUCTIVITY WEIGHT DUE TO DEVELOMENT
   |

TMPDEV(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
1,1,1,1,1,1,0.975,0.9,0.75,0.5,0)
   ~
   ~ |

MPWREX=TMPREX((1-MDPRNT/(JBSZMD-TSSZMD)))
   ~
   ~ MULTIPLIER TO PRODUCTIVITY WEIGHT DUE TO RESOURCE EXPENDITURE
(DIME
   |

TMPREX(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1,
1,1,1,1,1,1,0.975,0.9,0.75,0.5,0)
   ~
   ~ |

MDPNNT=TSKPRM/ASSPRD
   ~ MAN-DAYS
   ~ MAN DAYS PERCEIVED STILL NEEDED FOR NEW TASKS
   |

TMDPSN=MDPNNT+MDPNTS+MDPNRW
   ~ MAN-DAYS
   ~ TOTAL MAN DAYS PERCEIVED STILL NEEDED
   |

MDPNTS=TSTPRM/PRTPRD
   ~ MAN-DAYS
   ~ MAN DAYS PERCEIVED STILL NEEDED FOR TESTING
   |

TSTPRM=PJBSZ-CUMTKT
   ~ TASKS
   ~ TASKS REMAINING TO BE TESTED
   |

PRTPRD=SMOOTH((IF THEN ELSE(0 >= CUMTKT,PLTSPD,ACTSPD)),TSTSPD)
    ~ TASKS/MAN-DAY
    ~ PERCEIVED TESTING PRODUCTIVITY
    |

TSTSPD=50
    ~ DAYS
    ~ TIME TO SMOOTH TESTING PRODUCTIVITY
    |

PLTSPD=PJBSZ/TSSZMD
    ~ TASKS/MAN-DAY
    ~ PLANNED TESTING PRODUCTIVITY
    |

ACTSPD=CUMTKT/(CMTSMD+0.001)
    ~ TASKS/MAN-DAY
    ~ ACTUAL TESTING PRODUCTIVITY
    |

PMDSHR=TMDPSN-MDRM
    ~ MAN DAYS
    ~ PERCEIVED SHORTAGE IN MAN DAYS
    |

SHRRPT=PMDSHR-MDHDL
    ~ MAN-DAYS
    ~ SHORTAGE REPORTED
    |

MDRPTN=MDRM+SHRRPT
    ~ MAN DAYS
    ~ MAN DAYS REPORTED STILL NEEDED
    |

SCHPR=(TMDPSN-MDRM)/MDRM
    ~ DIMENSIONLESS
    ~ SCHEDULE PRESSURE
    |

PTRPTC=ACTIVE INITIAL(SMOOTH((100-(MDRPTN/JBSZMD)*100),RPTDLY),0)
    ~ Percent
    ~ Percent OF TASKS REPORTED COMPLETE
    |

51

RPTDLY=10
  ~ DAYS
  ~ REPORTING DELAY
  |

PDEVRC=ACTIVE INITIAL(SMOOTH(MAX((100-((MDRPTN-MDPNTS)/(JBSZMD-TSSZMD))*100),PDEVRC),RPTDLY),0)
  ~
  ~ Percent DEVELOPMENT PERCEIVED COMPLETE %
  |

UNDJTK=INTEG((-RTDSTK),RJBSZ-PJBSZ)
  ~ TASKS
  ~ UNDISCOVERED JOB TASKS
  |

RJBSZ=INITIAL(RJBDSI/DSIPTK)
  ~ TASKS
  ~ REAL JOB SIZE IN TASKS
  |

RTDSTK=UNDJTK*PUTDPD/100
  ~ TASKS/DAY
  ~ RATE OF DISCOVERING TASKS
  |

PUTDPD=TPUTDD(PJBPWK)
  ~ 1/DAY
  ~ PERCENT OF UNDISCOVERED TASKS DISCOVERED PER DAY
  |

TPUTDD(0,20,40,60,80,100,
0,0.4,2.5,5,10,100)
  ~
  ~ |

PJBPWK=(CMTKDV/PJBSZ)*100
  ~ Percent
  ~ Percent OF JOB PERCEIVED WORKED
  |

RTINCT=DELAY3(RTDSTK,DLINCT)
  ~ TASKS/DAY
  ~ RATE OF INCORPORATING DISCOVERED TASKS INTO PROJECT
  |

TKDSCV=INTEG((TIME STEP*(1/TIME STEP)*((-1)*TKDSCV+MAX((TKDSCV+TIME
STEP*(RTDSTK-
RTINCT)),0)))/TIME STEP,0)
~ TASKS
~ TASKS DISCOVERED
|

DLINCT=10
~ DAYS
~ AVERAGE DELAY IN INCORPORATING DISCOVERED TASKS
|

PJBSZ=INTEG(RTINCT,PJBDSI/DSIPTK)
~ TASKS
~ CURRENTLY PERCEIVED JOB SIZE
|

TSKPRM=PJBSZ-CMTKDV
~ TASKS
~ NEW TASKS PERCEIVED REMAINING
|

PSZDCT=TKDSCV/ASSPRD
~ MAN-DAYS
~ PERCEIVED SIZE OF DISCOVERED TASKS IN MAN DAYS
|

RSZDCT=PSZDCT/(MDPRNT+0.0001)
~ DIMENSIONLESS
~ RELATIVE SIZE OF DISCOVERED TASKS
|

FADHWO=TFAHWO(RSZDCT/(MSZTWO+0.001))
~
~ FRACTION OF ADDITIONAL TASKS ADDING TO MAN-DAYS
|

TFAHWO(0,0.2,0.4,0.6,0.8,1,1.2,1.4,1.6,1.8,2,
0,0,0,0,0,0,0.7,0.9,0.975,1,1)
~
~
|

53

MSZTWO=0.01
    ~
    ~ MAXIMUM RELATIVE SIZE OF ADDITIONS TOLERATED W/O ADDING TO
PROJECT'
    |

IRDVDT=(RTINCT/ASSPRD)*(FADHWO)
    ~
    ~ RATE OF INCREASE IN DEVELOPMENT MAN- DAYS DUE TO DISCOVERED
TASKS (M
    |

TSSZMD=INTEG((IRTSDT+(1/TIME STEP)*ARTJBM*IF THEN ELSE(FREFTS >=
0.9,1,0)),TSTMD)
    ~
    ~ PLANNED TESTING SIZE IN MAN-DAYS ... BEFORE WE START TESTING
    |

IRTSDT=(RTINCT/PRTPRD)*(FADHWO)
    ~ MD/D
    ~ RATE OF INCREASE IN TESTING MAN DAYS DUE TO DISCOVERED TASKS
    |

JBSZMD=TOTMD1
    ~ MAN DAYS
    ~ TOTAL JOB SIZE IN MAN DAYA
    |

ARTJBM=(MDRPTN+CUMMD-JBSZMD)/DAJBMD
    ~ MAN-DAYS/DAY
    ~ RATE OF ADJUSTING THE JOB SIZE IN MAN-DAYS
    |

DAJBMD=TDAJMD(TIMERM)
    ~ DAYS
    ~ DELAY IN ADJUSTING JOB'S SIZE IN MAN DAYS
    |

TDAJMD(0,20,
0.5,3)
    ~
    ~
    |

54

MDRM=MAX(0.0001,JBSZMD-CUMMD)

   ~

   ~

   |

{*************************
IV. PLANNING SUBSYSTEM
****
 MAN DAYS REMAINING
}
TIMEPR=MDRM/(WFS*ADMPPS)

  ~ DAYS

  ~ TIME PERCEIVED STILL REQUIRED

  |

INDCDT=Time+TIMEPR

   ~

  ~ INDICATED COMPLETION DATE

  |

SCHCDT=PROJDR

   ~

  ~ SCHEDULE COMPLETION DATE

  |

PROJDR=GAME(272)

  ~ DAYS

  ~ PROJECT DURATION

  |

SCHADT=TSHADT(TIMERM)

  ~ DAYS

  ~ SCHEDULE ADJUSTMENT TIME

  |

TSHADT(0,5,
0.5,5)

   ~

   ~

  |

TIMERM=MAX(SCHCDT-Time,0)

  ~ DAYS

  ~ TIME REMAINING

  |

WFINDC=(MDRM/(TIMERM+0.001))/ADMPPS
   ~ PEOPLE
   ~ INDICATED WORKFORCE
   |

WFNEED=MIN((WCWF*WFINDC+(1-WCWF)*TOTWF),WFINDC)
   ~ PEOPLE
   ~ WORKFORCE LEVEL NEEDED
   |

WCWF=MAX(WCWF1,WCWF2)
   ~ DIMENSIONLESS
   ~ WILLINGNESS TO CHANGE WORKFORCE LEVEL
   |

WCWF1=TWCWF1(TIMERM/WCWFTP)
   ~ DIMENSIONLESS
   ~ WILLINGNESS TO CHANGE WORKFORCE (1)
   |

TWCWF1(0,0.3,0.6,0.9,1.2,1.5,1.8,2.1,2.4,2.7,3,
0,0,0.1,0.4,0.85,1,1,1,1,1,1)
   ~
   ~ |

WCWFTP=IF THEN ELSE(TMPRMR = 0,HIREDY+ASIMDY,TMPRMR)
   ~
   ~ TIME PARAMETER DAYS
   |

TMPRMR=0
   ~
   ~ TIME PARAMETER (= HIREDY+ASIMDY) DAYS
   |

WCWF2=TWCWF2(SCHCDT/MXTLCD)
   ~ DIMENSIONLESS
   ~ WILLINGNESS TO CHANGE WF (2)
   |

TWCWF2(0.86,0.88,0.9,0.92,0.94,0.96,0.98,1,
0,0.1,0.2,0.35,0.6,0.7,0.77,0.8)
   ~
   ~
   |

MXTLCD=INITIAL(MXSCDX*TDEV)
    ~ DAYS
    ~ MAXIMUM TOLERABLE COMLETION DATE
    |

MXSCDX=1e+006
    ~ DIMENSIONLESS
    ~ MAX SCHEDULE COMPLETION DATE EXTENSION
    |

{*****************
V. INITIALIZATION
****

THE REAL JOB SIZE = 64,000 DSI

FROM BOEHM PAGE 90:
DISTRIBUTION OF EFFORT BY PHASE IS:
DESIGN (39%), PROGRAMMING (36%), INT TESTING (25%)

FROM BOEHM PAGE 64-65:
EFFORT = 2.4*(KDSI)**1.05
= 190 MM
= 190 * 19 = 3592 MAN-DAYS

DEVELOPMENT EFFORT = 75 %
= 2695 MAN DAYS
GROSS DEV PRODUCTIVITY = 64,000/2695 = 24 DSI/MD

SCHEDULE = 2.5 * (MM)**.38
= 18 MONTHS
= 348 DAYS
AVERAGE STAFF SIZE = 3592/348

= 10
GROSS PRODUCTIVITY INCORPORATES: DEV, FOR QA, & REWORKING
ASSUMING 25% OF EFFORT GOES INTO QA & REWORKING

25% OF 2695 MAN DAYS = 674 MAN DAYS
DEVELOPMENT PRODUCTIVITY = 64,000/(2695-674)

= 31 DSI/MAN-DAY
ASSUME LOSSES IN PRODUCTIVITY = 50 %
THEREFORE POTENTIAL PRODUCTIVITY = 31 * 2 = APPROX 60 DSI/MD
DEFINE 1 TASK = 60 DSI
}

57

DSIPTK=60

~
~ DSI PER TASK
|

RJBDSI=64000

~
~ REAL JOB SIZE IN DSI
|

UNDEST=0.33
~ FRACTION
~ TASKS UNDERESTIMATION FRACTION
|

PJBDSI=INITIAL(RJBDSI*(1-UNDEST))

~
~ PERCEIVED JOB SIZE IN DSI
|

TOTMD=INITIAL(MDSWCH*(((2.4*EXP(1.05*LN(PJBDSI/1000)))*19)*(1-UNDESM))+(1-
MDSWCH)
*TOTMD1)

~
~ TOTAL MAN DAYS
|

UNDESM=0
~ FRACTION
~ MAN-DAYS UNDERESTIMATION FRACTION
|

DEVMD=INITIAL(DEVPRT*TOTMD)

~
~ DEVELOPMENT MAN DAYS
|

MDSWCH=0

~
~ SWITCH 0 OR 1
|

TOTMD1INI=2359.6
    ~ MAN DAY
    ~   INITIAL ESTIMATED TOTAL MAN DAYS
    |

TOTMD1=GAME(TOTMD1INI)
    ~ MAN DAY
    ~   TOTAL MAN DAYS
    |

DEVPRT=0.8
    ~
    ~ Percent OF EFFORT ASSUMED NEEDED FOR DEVELOPMENT
    |

TSTMD=INITIAL((1-DEVPRT)*TOTMD)
    ~
    ~ TESTING MAN DAYS
    |

WFSTRT=WFS1
    ~ MEN
    ~ TEAM SIZE AT BEGINNING OF DESIGN
    |

INUDST=0.5
    ~ DIMENSIONLESS
    ~ INITIAL UNDERSTAFFING FACTOR
    |

TDEV=INITIAL(SCSWCH*((19*2.5*EXP(0.38*LN(TOTMD/19)))*SCHCOM)+(1-
SCSWCH)*TDEV1)
    ~ DAYS
    ~ TOTAL DEVELOPMENT TIME
    |

SCHCOM=1
    ~ DIMENSIONLESS
    ~ SCHEDULE COMPRESSION FACTOR
    |

SCSWCH=0
    ~
    ~ SWITCH 0 OR 1
    |

59

TDEV1=296.79

   ~

   ~ TIME TO DEVELOP

   |

TDEVINI=INTEGER(TDEV1)

   ~ DAYS

   ~   INITIAL TIME TO DEVELOP

   |

TEAMSZ=INITIAL((TOTMD/TDEV)/ADMPPS)

   ~

   ~ |

```
{**************************
VI. CONTROL STATEMENTS
**************************
}
```

TIME STEP=.5

   ~ DAYS

   ~ DT

   |

MAXLEN=1000

   ~

   ~

   |

SAVEPER=40

   ~ DAYS

   ~ CHANGED VALUE TO 40 DAYS VICE 10.

   |

FINAL TIME=IF THEN ELSE(PJBAWK >= 0.995,Time,MAXLEN)

   ~

   ~

   |

INITIAL TIME=0

   ~

   ~ Assumed with FINAL_TIME = LENGTH

   |

60

```
{****************************************************************
VII. EQUATIONS NECESSARY TO COMPLETE DYNAMO CONVERSION
****************************************************************
}

WFS2=WFS1
   ~ PEOPLE
   ~  WORK FORCE SOUGHT
   |

FRMPQ1=FRMPQA
   ~ PERSONS
   ~ FRACTION OF MANPOWER FOR Q1
   |

PRDFDS=PRERD/(MAX(PRTKDV/1000,0.01))
   ~ DEFECT/KDSI
   ~ PERIOD'S DEFECT DENSITY
   |

CMDSI=CMTKDV*DSIPTK
   ~ TASKS
   ~ CUMULATIVE TASKS DEVELOPED
   |

CRDVWF=TOTDMP-DMPQA
   ~ PERSONS
   ~ CURRENT DEVELOPMENT WORK FORCE
   |

CRQAWF=DMPQA
   ~ PERSONS
   ~ CURRENT QA WORK FORCE
   |

CRRWWF=DMPRW
   ~ PEOPLE
   ~  CURRENT REWORK WORK FORCE
   |

PRCMPL=(CMDSI/PJBSZT)*100
   ~ Percent
   ~ PERCENT COMPLETE
   |
```

61

RPPROD=PRDPRD*DSIPTK
    ~ DSI/MAN-DAY
    ~ REPORTED PRODUCTIVITY
    |

FNERG=INTEG((TIME STEP*ERRGRT*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME
STEP,0)
    ~ ERRORS
    ~ CUMULATIVE ERRORS GENERATED
    |

FNERD=INTEG((TIME STEP*ERRDRT*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME
STEP,0)
    ~ ERRORS
    ~ CUMULATIVE ERRORS DETECTED
    |

FNERES=FNERG-FNERD
    ~ ERRORS
    ~ ERRORS THAT ESCAPED QA
    |

FNPRDT=100*FNERD/MAX(1,FNERG)
    ~ Percent
    ~ PERCENT DETECTED
    |

FNQAMD=INTEG((TIME STEP*DMPQA*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME
STEP,0)
    ~ MAN-DAYS
    ~ CUMULATIVE QA MAN-DAYS
    |

FNTRMD=INTEG((TIME STEP*DMPTRN*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME
STEP,0)
    ~
    ~ CUMULATIVE TRAINING MAN-DAYS
    |

FNRWMD=INTEG((TIME STEP*DMPRW*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME
STEP,0)
    ~ MAN-DAYS
    ~ CUMULATIVE REWORK MAN-DAYS
    |

TM=Time
~
~
|

PJBSZT=PJBSZ*DSIPTK
~ DSI
~ PERCEIVED JOB SIZE IN LINES OF CODEPERCEIVED JOB SIZE IN DSI
|

IPRJSZ=INITIAL((RJBDSI)*(1-UNDEST))
~ DSI
~ INITIAL PROJECT SIZE IN DSI
|

FNCOST=INTEG((TIME STEP*TOTDMP*IF THEN ELSE(PJBAWK >= 0.995,0,1))/TIME STEP,0)
~ MAN-DAYS
~ FINAL COST IN MAN-DAYS
|

FNTIME=INTEG(IF THEN ELSE(PJBAWK >= 0.995,0,1),0)
~
~
|

FNERR=INTEG(((IF THEN ELSE(PJBAWK >= 0.995,FNERR,ALESER)/TIME STEP)-FNERR/TIME STEP),0)
~
~
|

PJBSZT KDSI=PJBSZT/1000
~ DSI
~ ESTIMATED SYSTEM SIZE (KDSI)
|

PRQAMD PERIOD=PRQAMD/((PRTKDV+0.01)/1000)
~ PERSON DAYS
~ QA PERSON DAYS PER KDSI DEVELOPED IN PERIOD
|

CUMMD TD=CUMMD-CMQAMD
~ PERSON DAYS
~ PROGRAMMING PERSON DAYS EXPENDED TO DATE
|

CMDSI KDSI=CMDSI/1000
   ~ TASKS
   ~ TOTAL KDSI COMPLETED
   |

CMERD KDSI=CMERD*(1000/(CMDSI+.01))
   ~ DEFECTS/KDSI
   ~ DEFECT DENSITY PER KDSI
   |

FRWFEX PCT=FRWFEX*100
   ~ DIMENSIOLESS
   ~ FRACTION OF WF THAT IS EXPERIENCED
   |

{**
VIII. VENSIM MACRO FOR DYNAMO FUNCTIONS
****

This macro definition is for a the PULSE function
commonly used in DYNAMO, but not directly supported
in Vensim.

NOTE: M_PULSE is equivalent to the DYNAMO function PULSE
   M_PULSE takes Time as an argument
   SAMPLE takes Time and TIME STEP as arguments
}

:MACRO: M PULSE(time,height,width,first,intvl)
M PULSE = height* PULSE(pulse_start,width)
   ~ height
   ~ Note that the pulse function in Vensim is different from the pulse
   function in DYNAMO. The argument time needs to be added to get
   repeated pulses - things may still be misaligned.
   |

pulse_start = IF THEN ELSE(time < first+intvl,first,first + (QUANTUM((time-first)/intvl,1))*intvl)
   ~ time
   ~ The pulse start moves forward over time. It makes no sense to have
   width > interval, and interval = 0 will cause an error.
   |

:END OF MACRO:

64

:MACRO: SAMPLE(Time,TIME STEP,INITIAL TIME,X,INTVL,ISAM)
SAMPLE=SAMPLE IF TRUE(Time > next_time,X,X)
    ~
    ~
    |

next_time=INTEG(IF THEN ELSE(Time>next_time,INTVL/TIME STEP,0),
 INITIAL TIME-TIME STEP/2)
    ~ Time
    ~
    |

:END OF MACRO:

PRQAMD=INTEG((DMPQA-(PRQAMD/TIME STEP)*M PULSE(Time,1,TIME STEP,TIME
STEP,40)),0)
    ~ PERSON-DAYS
    ~ QA PERSON-DAYS IN PERIOD
    |

PRERD=INTEG((ERRDRT-(PRERD/TIME STEP)*M PULSE(Time,1,TIME STEP,TIME
STEP,40)),0)
    ~ ERRORS
    ~ DETECTED ERRORS DURING PERIOD
    |

TMSTOP=IF THEN ELSE(PJBAWK*M PULSE(Time,1,TIME STEP,TIME STEP,40) >=
0.995,Time,MAXLEN)
    ~
    ~ FINAL TIME
    |

PRTKDV=INTEG((SDVRT*DSIPTK-(PRTKDV/TIME STEP)*M PULSE(Time,1,TIME
STEP,TIME STEP,40)),0.1)
    ~
    ~ TASKS DEVELOPED DURING 40 DAY PERIOD
    |

APPENDIX B. VENSIM CUSTOM DEFINITION FILE

:SCREEN WELCOME
SCREENFONT,Times New Roman|12||0-0-0
TEXTONLY,"Software Management Flight Simulator",0,15,100,0,C|Arial|20|B|255-25-0|,
TEXTONLY,"",0,38,100,0,C05,
TEXTONLY,"",0,41,100,0,C05
TEXTONLY,"",0,44,100,0,C05
TEXTONLY,"Press any Key to Continue",0,60,100,0,C,
ANYKEY,"",0,0,0,0,0,,,INTRO

:SCREEN INTRO
TEXTONLY,"Software Management Flight Simulator",0,5,100,0,C|Arial|20|B|255-25-0|,
TEXTONLY, "You are not allowed to discuss this exercise with anyone other than the labattendant.",5,15,100,0,L
TEXTONLY, "Please refrain from discussing this with members in the other class until they havecompleted",5,20,100,0,L
TEXTONLY, "the exercise.",5,25,100,0,L
TEXTONLY, "",5,30,100,0,L
TEXTONLY, "The system will show you the size of the initial core team of software developers who have just",5,35,100,0,L
TEXTONLY, "completed the requirements/design specifications. The system will then advance to the",5,40,100,0,L
TEXTONLY, "programming phase where you will simulate the first 40 working day time period. You will",5,45,100,0,L
TEXTONLY, "be allowed to view the various reports and graphs and then update your estimates for the",5,50,100,0,L
TEXTONLY, "project cost and duration and change your staffing levels.",5,55,100,0,L
TEXTONLY, "",5,60,100,0,L
TEXTONLY, "Record your decisions for each interval on the documentation sheet provided before",5,65,100,0,L
TEXTONLY, "proceeding to the next level.",5,70,100,0,L
TEXTONLY, "",5,75,100,0,L
TEXTONLY, "THE LAB ATTENDANT MUST VERIFY YOUR FINAL RESULTS. GOOD LUCK!",5,80,100,0,L
TEXTONLY,"Press any Key to Continue",0,95,100,0,C,
ANYKEY,"",0,0,0,0,0,,,MAIN

:SCREEN MAIN
TEXTONLY,"Software Management Flight Simulator",0,5,100,0,C|Arial|20|B|255-0-0
TEXTMENU,"Play New Game",5,15,0,0,L,Pp,SIMULATE>RUNNAME|?NAME FOR NEW
GAME OUTPUT (NOT BASE!!),STARTGAME
TEXTMENU,"Go back to the introductory material",5,20,0,0,L,GgIi,,INTRO
TEXTMENU,"Overview",5,25,0,0,L,Rr,,MODEL_OVERVIEW
TEXTMENU,"Display Models",5,30,0,0,L,Rr,,MODELS
TEXTMENU,"Analyze Previous Run Scenario",5,35,0,0,L,Rr,,ANALYZE
TEXTMENU,"Exit the System",5,40,0,0,L,EeXx,SPECIAL>ASKYESNO|Do you really want to
exit the simulator?&MENU>EXIT,

:SCREEN STARTGAME
COMMAND,"",0,0,0,0,,,SPECIAL>LOADMODEL|snsw15.vmf
COMMAND,"",0,0,0,0,,,SPECIAL>READCUSTOM|snsw15.vgd
COMMAND,"",0,0,0,0,,,SPECIAL>CLEARRUNS
COMMAND,"",0,0,0,0,,,SIMULATE>BASED|
COMMAND,"",0,0,0,0,,,SIMULATE>RESUME|10
COMMAND,"",0,0,0,0,,,GAME>GAMEINTERVAL|40
COMMAND,"",0,0,0,0,,,MENU>GAME|O
COMMAND,"",0,0,0,0,,,SIMULATE>CHGFILE|
COMMAND,"",0,0,0,0,,,SIMULATE>BASED|
COMMAND,"",0,0,0,0,,,SIMULATE>RESUME|O
CLOSESCREEN,"",0,0,0,0,,,,INITIAL_ESTIMATE

:SCREEN INITIAL_ESTIMATE
TEXTONLY,"Software Management Flight Simulator",0,5,100,0,C|Arial|20|B|255-25-0|,
TEXTONLY,"The initial core team of software developers have just completed the requirements
and",5,15,100,0,L
TEXTONLY,"design specifications. Your task is to take over as manager of the programming
phase.",5,20,100,0,L
TEXTONLY,"Shown below are the Initial Project Estimates:",5,25,0,0,L
TEXTONLY,"System Size in DSI",22,35,0,0,L
SHOWVAR,"IPRJSZ",68,35,0,0,R
TEXTONLY,"Cost of Programming Phase",22,40,0,0,L
SHOWVAR,"TOTMD1",68,40,0,0,R
TEXTONLY,"Duration of Programming Phase",22,45,0,0,L
SHOWVAR,"TDEV",68,45,0,0,R
TEXTONLY,"Initial Development Team",22,50,0,0,L
SHOWVAR,"WFS1",68,50,0,0,R
TEXTONLY,"Pct of Staff Allocated to QA",22,55,0,0,L
SHOWVAR,"FRMPQA",68,55,0,0,R
TEXTONLY,"NEW_TOOL's estimate for the percent of the total staff to allocate to QA is
shown above.",5,65,100,0,L

TEXTONLY,"Remember, NEW_TOOL has not yet been calibrated to your environment. This estimate is",5,70,100,0,L
TEXTONLY,"merely illustrative. It may or may not be appropriate for your unique project. At this",5,75,100,0,L
TEXTONLY,"point, you need to make two decisions based on this information.",5,80,100,0,L
TEXTONLY,"Press Any Key to Continue",0,95,100,0,C,
ANYKEY,"",0,0,0,0,0,,,FIRST_DECISION

:SCREEN FIRST_DECISION
TEXTONLY,"Initial Programming Phase Decisions",0,5,100,0,C|Arial|20|B|255-25-0|,
TEXTONLY, "FIRST DECISION: Determine the total staff level for the Programming Phase.",0,20,100,0,C
TEXTONLY,"Staffing Level",35,30,0,0,L
MODVAR,"WFS1",57,30,7,5,,[0|]
TEXTONLY, "SECOND DECISION: Determine the percentage of personnel allocated to Quality Assurance.",0,45,100,0,C
TEXTONLY,"Pct Alloc to QA",35,55,0,0,L
MODVAR,"FRMPQA",57,55,7,5,,[0|100]
TEXTONLY, "IMPORTANT!!",0,70,100,0,C
TEXTONLY, "This is your final opportunity to change these values.",0,78,100,0,C
BUTTON,"Advance Time",38,90,25,0,,,GAME>GAMEON,HOME_BASE

:SCREEN HOME_BASE
TEXTONLY,"Software Management Flight Simulator",0,5,100,0,C|Arial|20|B|255-25-0|,
RECTANGLE,"",2,15,40,30
TEXTONLY,"INITIAL PROJECT ESTIMATES",5,16,0,0,L
TEXTONLY,"System Size in DSI",3,23,0,0,L
SHOWVAR,"IPRJSZ",34,23,7,3,R
TEXTONLY,"Cost of Programming Phase",3,28,0,0,L
SHOWVAR,"TOTMD1",34,28,7,3,R
TEXTONLY,"Duration of Programming Phase",3,33,0,0,L
SHOWVAR,"TDEV",34,33,7,3,R
TEXTONLY,"Initial Development Team",3,38,0,0,L
SHOWVAR,"WFS1",34,38,7,3,R
RECTANGLE,"",2,47,40,40
TEXTONLY,"INPUT VARIABLE VALUES",7,48,0,0,L
TEXTONLY,"Project Cost",3,55,0,0,L
MODVAR,"TOTMD1",32,55,8,5,R,[0|]
TEXTONLY,"Project Duration",3,61,0,0,L
MODVAR,"PROJDR",32,61,8,5,R,[0|]
TEXTONLY,"Staffing Level",3,67,0,0,L
MODVAR,"WFS1",32,67,8,5,R,[0|]
TEXTONLY,"Pct Alloc to QA",3,73,0,0,L
MODVAR,"FRMPQA",32,73,8,5,R,[0|100]

BUTTON,"Explanation",9,79,25,0,L,,CUSTOM>EXP1
BUTTON,"Project Status Report",45,37,25,0,L,,CUSTOM>RPT2
BUTTON,"Project Status Graphs",72,37,25,0,L,,,PROJ_STATUS_GRAPH1
BUTTON,"Staffing Report",45,47,25,0,L,,CUSTOM>RPT1
BUTTON,"Staffing Graph",72,47,25,0,L,,,STAFFING_GRAPH
BUTTON,"Defect Report",45,57,25,0,L,,CUSTOM>RPT3
BUTTON,"Defect Graph",72,57,25,0,L,,,DEFECT_GRAPH
BUTTON,"Analyze Scenario",45,67,25,0,L,,,ANALYZE
BUTTON,"End Simulation",72,67,25,0,L,,SPECIAL>ASKYESNO|Do you really want to end
this simulation?&GAME>ENDGAME,MAIN
BUTTON,"Advance Time",38,93,25,0,L,,GAME>GAMEON&BRANCH>END
BRANCH,"END",0,0,0,0,,,IFTHENELSE&TEST>PJBAWK<=.995&SPECIAL>REFRESH&G
AME>ENDGAME

:SCREEN PROJ_STATUS_GRAPH1
TOOL,"GR1",0,0,100,90,,,CUSTOM>STATUS_GRAPH1
BUTTON,"Go Back",5,93,25,0,L,,,HOME_BASE
BUTTON,"Unstack Graphs",37,93,25,0,L,,,UNSTACKED_MENU
BUTTON,"Print Current View",70,93,25,0,L,,PRINT>GR1

:SCREEN UNSTACKED_MENU
TEXTONLY,"Listed below are the three individual graphs which are plotted on
the",0,5,100,0,C|Times New Roman|14|
TEXTONLY,"Project Status Graph. Select the desired graph or table of numeric
values.",0,10,100,0,C|Times New Roman|14|
BUTTON,"Pgm Cost Graph",2,86,20,0,L,,,UNSTACKED_PGM_COST
BUTTON,"Pgm Cost Table",2,93,20,0,L,,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,,,UNSTACKED_SYS_SIZE
BUTTON,"Sys Size Table",27,93,20,0,L,,,SYS_SIZE_TABLE
BUTTON,"Staffing Graph",52,86,20,0,L,,,UNSTACKED_TOT_STAFF
BUTTON,"Staffing Table",52,93,20,0,L,,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,,PRINT>GR1

:SCREEN UNSTACKED_PGM_COST
TOOL,"GR1A1",0,0,100,80,,,CUSTOM>UNSTACKED_2A
BUTTON,"Pgm Cost Graph",2,86,20,0,L,,,UNSTACKED_PGM_COST
BUTTON,"Pgm Cost Table",2,93,20,0,L,,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,,,UNSTACKED_SYS_SIZE
BUTTON,"Sys Size Table",27,93,20,0,L,,,SYS_SIZE_TABLE
BUTTON,"Staffing Graph",52,86,20,0,L,,,UNSTACKED_TOT_STAFF

BUTTON,"Staffing Table",52,93,20,0,L,,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,,PRINT>GR1A1

:SCREEN PGM_COST_TABLE
TEXTONLY,"Estimated Programming Cost",0,5,100,0,C|Times New Roman|16|
COMMAND,"",0,0,0,0,,,SPECIAL>SETWBITEM|JBSZMD
TOOL,"GR1A2",0,30,100,40,,,WORKBENCH>CAUSES TAB
BUTTON,"Pgm Cost Graph",2,86,20,0,L,,,UNSTACKED_PGM_COST
BUTTON,"Pgm Cost Table",2,93,20,0,L,,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,,,UNSTACKED_SYS_SIZE
BUTTON,"Sys Size Table",27,93,20,0,L,,,SYS_SIZE_TABLE
BUTTON,"Staffing Graph",52,86,20,0,L,,,UNSTACKED_TOT_STAFF
BUTTON,"Staffing Table",52,93,20,0,L,,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,,PRINT>GR1A2

:SCREEN UNSTACKED_SYS_SIZE
TOOL,"GR2A1",0,0,100,80,,,CUSTOM>UNSTACKED_2B
BUTTON,"Pgm Cost Graph",2,86,20,0,L,,,UNSTACKED_PGM_COST
BUTTON,"Pgm Cost Table",2,93,20,0,L,,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,,,UNSTACKED_SYS_SIZE
BUTTON,"Sys Size Table",27,93,20,0,L,,,SYS_SIZE_TABLE
BUTTON,"Staffing Graph",52,86,20,0,L,,,UNSTACKED_TOT_STAFF
BUTTON,"Staffing Table",52,93,20,0,L,,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,,PRINT>GR2A1

:SCREEN SYS_SIZE_TABLE
TEXTONLY,"Estimated System Size",0,5,100,0,C|Times New Roman|16|
COMMAND,"",0,0,0,0,,,SPECIAL>SETWBITEM|PJBSZT KDSI
TOOL,"GR2A2",0,30,100,40,,,WORKBENCH>CAUSES TAB
BUTTON,"Pgm Cost Graph",2,86,20,0,L,,,UNSTACKED_PGM_COST
BUTTON,"Pgm Cost Table",2,93,20,0,L,,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,,,UNSTACKED_SYS_SIZE
BUTTON,"Sys Size Table",27,93,20,0,L,,,SYS_SIZE_TABLE
BUTTON,"Staffing Graph",52,86,20,0,L,,,UNSTACKED_TOT_STAFF
BUTTON,"Staffing Table",52,93,20,0,L,,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,,PRINT>GR2A2

```
:SCREEN UNSTACKED_TOT_STAFF
TOOL,"GR3A1",0,0,100,80,,,CUSTOM>UNSTACKED_2C
BUTTON,"Pgm Cost Graph",2,86,20,0,L,,,UNSTACKED_PGM_COST
BUTTON,"Pgm Cost Table",2,93,20,0,L,,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,,,UNSTACKED_SYS_SIZE
BUTTON,"Sys Size Table",27,93,20,0,L,,,SYS_SIZE_TABLE
BUTTON,"Staffing Graph",52,86,20,0,L,,,UNSTACKED_TOT_STAFF
BUTTON,"Staffing Table",52,93,20,0,L,,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,,PRINT>GR3A1


:SCREEN TOT_STAFF_TABLE
TEXTONLY,"Total Staffing Level",0,5,100,0,C|Times New Roman|16|
COMMAND,"",0,0,0,0,,,SPECIAL>SETWBITEM|FTEQWF
TOOL,"GR3A2",0,30,100,40,,,WORKBENCH>CAUSES TAB
BUTTON,"Pgm Cost Graph",2,86,20,0,L,,,UNSTACKED_PGM_COST
BUTTON,"Pgm Cost Table",2,93,20,0,L,,,PGM_COST_TABLE
BUTTON,"Sys Size Graph",27,86,20,0,L,,,UNSTACKED_SYS_SIZE
BUTTON,"Sys Size Table",27,93,20,0,L,,,SYS_SIZE_TABLE
BUTTON,"Staffing Graph",52,86,20,0,L,,,UNSTACKED_TOT_STAFF
BUTTON,"Staffing Table",52,93,20,0,L,,,TOT_STAFF_TABLE
BUTTON,"Go Back",78,86,20,0,L,,,PROJ_STATUS_GRAPH1
BUTTON,"Print Current View",78,93,20,0,L,,PRINT>GR3A2


:SCREEN DEFECT_GRAPH
TOOL,"",0,0,100,90,,,CUSTOM>DEFECTS
BUTTON,"Go Back",5,93,25,0,L,,,HOME_BASE
BUTTON,"View Table",37,93,25,0,L,,,DEFECT_TABLE
BUTTON,"Print Current View",70,93,25,0,L,,PRINT>DEFECT


:SCREEN DEFECT_TABLE
TEXTONLY,"Total Defects",0,5,100,0,C|Times New Roman|16|
COMMAND,"",0,0,0,0,,,SPECIAL>SETWBITEM|PRQAMD PERIOD
TOOL,"GR4",0,30,100,40,,,WORKBENCH>CAUSES TAB
COMMAND,"",0,0,0,0,,,SPECIAL>SETWBITEM|PRDFDS
TOOL,"GR4",0,70,100,40,,,WORKBENCH>CAUSES TAB
BUTTON,"Go Back",17,93,25,0,L,,,DEFECT_GRAPH
BUTTON,"Print Current View",60,93,25,0,L,,PRINT>GR4


:SCREEN STAFFING_GRAPH
TOOL,"",0,0,100,90,,,CUSTOM>STAFFING_LEVEL
BUTTON,"Go Back",5,93,25,0,L,,,HOME_BASE
```

```
BUTTON,"View Table",37,93,25,0,L,,,STAFFING_TABLE
BUTTON,"Print Current View",70,93,25,0,L,,PRINT>STAFFING_LEVEL


:SCREEN STAFFING_TABLE
TEXTONLY,"Total Staff Composition",0,5,100,0,C|Times New Roman|16|
COMMAND,"",0,0,0,0,,,SPECIAL>SETWBITEM|FTEQWF
TOOL,"GR5",0,20,100,40,,,WORKBENCH>CAUSES TAB
COMMAND,"",0,0,0,0,,,SPECIAL>SETWBITEM|CRQAWF
TOOL,"GR5",0,55,100,40,,,WORKBENCH>CAUSES TAB
COMMAND,"",0,0,0,0,,,SPECIAL>SETWBITEM|CRDVWF
TOOL,"GR5",0,85,100,30,,,WORKBENCH>CAUSES TAB
BUTTON,"Go Back",17,93,25,0,L,,,STAFFING_GRAPH
BUTTON,"Print Current View",60,93,25,0,L,,PRINT>GR5
```

APPENDIX C. VENSIM GRAPH DEFINITION FILE

:REPORT COMM4
:LOCATION -1,-1
:SIZE -1,-1

An initial core team of 2 software developers have just completed the requirements and design specifications.
Your task is to take over as manager of the programming phase. At this point, you need to make 2 decisions:

1. The total staff level for the programming phase.
2. The percent of this staff to allocate to Quality Assurance.

YOUR FIRST DECISION: Enter the Total Staff Level

merely illustrative. It may or may not be appropriate for your unique project.

YOUR SECOND DECISION: Allocate 10 Percent of your staff to QA or modify the amount
(Range 0-100) and press
ALT+R to run the scenerio.
:END-OF-REPORT

:REPORT EXP1
:SIZE 40,10
:TITLE Input Variable Values
 Project Cost: Enter your estimates for total Project Cost in Person-Days.

 Project Duration: Enter your updated estimate for the Project Duration in days.

 Staffing Level: Enter your total requested Staffing Level.

 Pct Alloc to QA: Enter the desired percent of personnel allocated to Quality
    Assurance as a number from 0 to 100.
:END-OF-REPORT

:REPORT RPT1
:SIZE 35,10
:TITLE Staffing Report

At time = \TM/

Current Total Staff Size   \FTEQWF/ People
   Staff Allocated to Programming   \CRDVWF/ People
   Staff Allocated to QA   \CRQAWF/ People

Percent of Workforce that is Experienced   \FRWFEX PCT/ Percent
:END-OF-REPORT

:REPORT RPT2
:SIZE 45,20
:TITLE Project Status Report

At Time = \TM/

UPDATED ESTIMATES
   New Est of System Size due to changes in requirements \PJBSZT/ DSI
   Your Last Est of Programming Phase Cost   \JBSZMD/ Person-Days
   Your Last Est of Programming Phase Duration   \SCHCDT/ Days
   Time Remaining   \TIMERM/ Days

REPORTED PROGRESS
   Percent DSI Reported Complete   \PRCMPL/ Percent
   Total DSI Reported Complete to Date   \CMDSI/ DSI
   Total Person-Days Expended to Date   \CUMMD/ Person-Days
   Reported Productivity   \RPPROD/ DSI/Person-Days

:END-OF-REPORT

:REPORT RPT3
:SIZE 45,20
:TITLE Defect Report

At Time = \TM/

CUMMULATIVE STATISTICS FROM START OF PROJECT
   TOTAL Person Days Expended to Date   \CUMMD/ Person-Days
   Programming Person Days Expended to Date   \CUMMD TD/ Person-Days
   QA Person-Days Expended to Date   CMQAMD/ Person-Days

TOTAL Defects Detected   \CMERD/ Defects
TOTAL KDSI Completed   \CMDSI KDSI/ KDSI
Defect Density   \CMERD KDSI/ Defects/KDSI

STATISTICS FOR THE LAST 40 DAY PERIOD ONLY
   QA Person Days Expended   \PRQAMD/   Person-Days
   Defects Detected   \PRERD/   Defects
   Density of Defects Detected   \PRDFDS/   Defects/KDSI

:END-OF-REPORT

:GRAPH WIP1
:TITLE Progress and Work Effort
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 300
:WIP
:SCALE
:VAR PTKTST|Fraction of task tested
:UNITS Fraction
:Y-MIN 0
:Y-MAX 1
:SCALE
:VAR PTRPTC|Percentage of task reported complete
:UNITS %
:Y-MIN 0
:Y-MAX 100
:SCALE
:VAR CMDVMD
:UNITS Person-Days
:Y-MIN 0
:Y-MAX 2000
:GRAPH TOT_COST2
:TITLE TOT_COST2
:SCALE
:VAR total cum cost
:Y-MIN 0
:Y-MAX 2e+008
:SCALE
:VAR REP FRAC COMPLETE[DESIGN]
:VAR REP FRAC COMPLETE[PRODUCE]
:VAR REP FRAC COMPLETE[TEST]
:Y-MIN 0
:Y-MAX 1

:GRAPH STATUS_GRAPH1
:TITLE Project Size and Status Graph
:X-LABEL Time in Days

77

:X-MIN 0
:X-MAX 600
:VAR JBSZMD|Estimated Programming Cost
:UNITS Person-Days
:Y-MIN 0
:Y-MAX 4000
:VAR PJBSZT KDSI|Estimated System Size
:UNITS KDSI
:Y-MIN 0
:Y-MAX 64
:VAR FTEQWF|Total Staff
:UNITS Persons
:Y-MIN 0
:Y-MAX 24

:GRAPH UNSTACKED_2A
:TITLE Estimated Programming Cost
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:VAR JBSZMD
:UNITS Person-Days
:Y-MIN 0
:Y-MAX 4000

:GRAPH UNSTACKED_2B
:TITLE Estimated System Size
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:VAR PJBSZT KDSI
:UNITS KDSI
:Y-MIN 0
:Y-MAX 64

:GRAPH UNSTACKED_2C
:TITLE Total Staffing Level
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:SCALE
:VAR FTEQWF
:UNITS Persons

:Y-MIN 0
:Y-MAX 24

:GRAPH DEFECTS
:TITLE Total Defects
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:WIP
:VAR PRQAMD PERIOD|QA Person-Days per KDSI,PRDFDS|Defects Detected per KDSI
:Y-MIN 0
:Y-MAX 80
:SCALE

:GRAPH STAFFING_LEVEL
:TITLE Total Staff Composition
:X-LABEL Time in Days
:X-MIN 0
:X-MAX 600
:VAR FTEQWF|Total Staff,CRDVWF|Pgm Staff,CRQAWF|QA Staff
:UNITS Persons
:Y-MIN 0
:Y-MAX 24

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 013 2
   Naval Postgraduate School
   Monterey, California 93943-5101

3. Director, Training and Education 1
   MCCDC, Code C46
   1019 Elliot Rd.
   Quantico, Virginia 22134-5027

4. Director, Information Systems (OP-945) 1
   Office of the Chief of Naval Operations
   Naval Department
   Washington, D. C. 20350-0001

5. Dr. Tarek Abdel-Hamid, Code SM/Ah..5
   Department of Systems Management
   Naval Postgraduate School
   Monterey, California 93943

6. Dr. Hemant Bhargava, Code SM/BH 1
   Department of Systems Management
   Naval Postgraduate School
   Monterey, California 93943

7. Captain Richard D. Davis, Jr. 1
   5 Wren Way Court
   Stafford, Virginia 22554